CERTIFIED DATA ATTRIBUTION AFTER MODEL RESELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Data deletion can change the configuration selected by validation as well as its fitted parameters. We study attribution to the resulting retrain-and-reselect procedure over a finite menu of strongly convex learners. A selected output can be certified without identifying the winning configuration. We characterize the feasible attribution set under interval information with deterministic ties and derive its minimax absolute-error radius. Residual regions and validation-conditioned prediction bounds give an adaptive procedure whose refinement bound depends on disagreement among near-optimal candidates. A three-observation ridge construction has a nonzero deletion effect and an output-certification cost independent of an arbitrarily small validation gap under a shared damped-Newton schedule. Full Newton steps remove this separation, showing that refinement granularity matters. Across 4,800 classification queries, reselection occurs in 12.3–19.1% of queries by setting. Output stopping saves no refinements at a joint 10^{-3} signed-logit tolerance. A complete probability-output sweep yields 1.5–12.4% fewer refinements than equally certified winner identification for individual targets, with substantially smaller gains for simultaneous outputs. Additional conditioning contributes little and does not improve measured latency. The results characterize when the output, tolerance, and solver permit computational savings.

1 INTRODUCTION

Removing a training source can change both a model's fitted parameters and the configuration selected by validation. Attribution at the original regularization strength holds that configuration fixed, even when its parameters are refitted exactly. When the training procedure repeats model selection after deletion, the relevant counterfactual is the output of that complete procedure on the modified data.

Influence functions approximate changes within a fitted configuration (Koh & Liang, 2017), and group-deletion analyses distinguish magnitude error from useful rankings (Koh et al., 2019). General learning-procedure targets are also established. Data Shapley and datamodels permit such targets (Ghorbani & Zou, 2019; Ilyas et al., 2022), and influence-based model selection predates modern attribution methods (Debruyne et al., 2008). The question here is computational. How much of a finite candidate menu must be refitted to certify a selected prediction after a specified deletion?

Identifying the validation winner is sufficient but not always necessary. Nearly tied configurations may make almost identical predictions on one queried example and substantially different predictions on another. A small validation gap alone therefore does not determine the difficulty of attribution. What matters is the range of selected outputs compatible with the available information. Nor does a small deleted fraction ensure a small effect from changing the winner, as the bounded ridge example below shows.

We develop *switch-aware attribution* (SWA) to certify this range. Our information-level result characterizes the possible attributions under score and output intervals with deterministic ties. For strongly convex objectives, classical residual regions and validation pruning (Suzuki et al., 2014) provide these intervals. The established dome-support formula (Xiang et al., 2016) restricts predictions further under the premise that a candidate wins. We prove safety when such conditional information is accumulated, and bound synchronous refinement work through near-optimal prediction disagreement. A finite-data ridge construction exhibits the separation using actual residual regions and a

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

nonzero deletion effect. The comparison concerns two specified refinement procedures, not arbitrary algorithms with data access.

The experiments distinguish attribution accuracy, refinement savings, and elapsed time. Across 4,800 deletion queries, reselection changes attribution, but the three stopping variants use identical work at stringent joint-logit accuracy. We then vary probability tolerance, regularization range, menu density, and the number of queried outputs without selecting deletions by their observed gains. Certifying each of the 16 targets separately saves a modest number of refinements; most of that saving disappears when they must be certified together. The study also evaluates sign and intervention-order decisions, point approximations at equal refinement counts, and counterbalanced query times. A larger controlled simulation tests sample-size scaling, not external-data generalization. These comparisons show why a valid certificate and tighter prediction bounds need not yield a faster implementation.

2 THE COUNTERFACTUAL AND THE ROLE OF RESELECTION

Let $D = (z_1, \dots, z_n)$ be a training sample, $G \subset \{1, \dots, n\}$ a deletion group, $s = |G| < n$, and $D^- = D \setminus G$. The ordered finite menu is $\mathcal{H} = \{1, \dots, K\}$. Candidate h has parameters in $\mathbb{R}^{p_h}$ and minimizes

$$R_h(\theta; S) = \frac{1}{|S|} \sum_{i \in S} \ell_h(z_i, \theta) + \frac{\lambda_h}{2} \|\theta\|^2, \qquad \theta_h(S) = \arg\min_{\theta} R_h(\theta; S). \tag{1}$$

The losses are convex and differentiable and $\lambda_h > 0$. The residual-enclosure results more generally require a differentiable μ_h-strongly convex objective with known $\mu_h > 0$. Newton refinement additionally requires twice continuously differentiable training objectives. The local rate bounds require a Lipschitz Hessian on the relevant step segments (Appendix G). A fixed validation sample defines a score V_h, and g_h is the scalar output being attributed. Define

$$h^*(S) = \min \arg\min_{h \in \mathcal{H}} V_h(\theta_h(S)), \qquad F(S) = g_{h^*(S)}(\theta_{h^*(S)}(S)), \qquad \Delta_G = F(D^-) - F(D). \tag{2}$$

The outer minimum fixes ties by menu order. The validation sample, menu, features, normalization, and any upstream randomness are held fixed across the intervention. In particular, the denominator in (1) changes from n to $n - s$, whereas the regularization coefficient does not. Retraining a feature extractor or recomputing preprocessing defines a different counterfactual.

Write $h_0 = h^*(D)$ and $h_* = h^*(D^-)$. The identity

$$\begin{aligned} \Delta_G = {} & \underbrace{g_{h_0}(\theta_{h_0}(D^-)) - g_{h_0}(\theta_{h_0}(D))}_{\Delta_G^{\text{fixed}}} \\ & + \underbrace{g_{h_*}(\theta_{h_*}(D^-)) - g_{h_0}(\theta_{h_0}(D^-))}_{\text{reselection contribution}} \end{aligned} \tag{3}$$

separates approximation error from conditioning on the wrong configuration. The second term vanishes when $h_* = h_0$, but need not be small when s/n is small.

Proposition 1 (A vanishing deletion fraction need not imply a vanishing selected effect)**.** *There is a one-dimensional ridge problem with bounded observations, regularization menu $\{1, 3\}$, and a fixed validation observation such that deleting one of $n \geq 3$ training observations gives*

$$\Delta_G = \frac{1}{4} - \frac{1}{4(n-1)} - \frac{1}{8n} > 0, \qquad \Delta_G^{\text{fixed}} = -\frac{1}{8(n-1)} - \frac{1}{8n} < 0.$$

Thus $\Delta_G \to 1/4$ while $\Delta_G^{\text{fixed}} \to 0$.

The construction in Appendix A approaches a selection boundary. It is a worst-case result, not a claim about how often this behavior occurs. A uniform guarantee based only on deletion fraction therefore needs additional assumptions on the selection margin or output agreement.

3 CERTIFICATES FOR THE SELECTED OUTPUT

For a fixed deletion, abbreviate $v_h = V_h(\theta_h(D^-))$ and $f_h = g_h(\theta_h(D^-))$. Suppose the computation yields valid closed intervals

$$v_h \in [L_h, U_h], \qquad f_h \in [a_h, b_h], \qquad F(D) \in [\underline{F}_0, \overline{F}_0]. \tag{4}$$

These are deterministic computational enclosures, not confidence intervals for a population quantity. Let $u = \min_j U_j$ and $\mathcal{S} = \{h : L_h \le u\}$.

Theorem 2 (Selected-output enclosure and interval-information sharpness)**.** *The selected candidate belongs to $\mathcal{S}$, and*

$$\Delta_G \in I_G = \left[\min_{h\in\mathcal{S}} a_h - \overline{F}_0, \max_{h\in\mathcal{S}} b_h - \underline{F}_0\right]. \tag{5}$$

With menu-order ties, the exact feasible-winner set under independent score intervals is

$$\mathcal{S}_{\text{tie}} = \{h : L_h \le U_j \text{ for every } j, \quad L_h < U_j \text{ for every } j < h\}. \tag{6}$$

Under independent score, output, and baseline intervals, the exact feasible attribution set is $\bigcup_{h\in\mathcal{S}_{\text{tie}}}[a_h - \overline{F}_0, b_h - \underline{F}_0]$. Its convex hull has minimax absolute-error radius equal to half its width.

The containment follows from $L_{h_*} \le v_{h_*} \le v_j \le U_j$ for every j. For feasibility in (6), choose $v_h = L_h$ and $v_j = U_j$ for $j \ne h$. The sharpness claim is relative to these rectangular information sets. It does not assert that arbitrary endpoint combinations arise from a common training dataset. A complete proof is given in Appendix B.

A midpoint of (5) has error at most ε whenever $\operatorname{diam}(I_G) \le 2\varepsilon$. The candidate need not be unique. For multiple outputs, we apply the construction coordinatewise and stop when the maximum coordinate width is at most 2ε, obtaining an ℓ_∞ guarantee. A positive lower endpoint or negative upper endpoint additionally certifies the sign.

Two candidates, two different stopping decisions. Suppose both validation intervals are $[0.20, 0.21]$ and the baseline is known to be zero. Output intervals $[0.49, 0.51]$ and $[0.50, 0.52]$ give the enclosure $[0.49, 0.52]$, so $\varepsilon = 0.02$ permits stopping without identifying a winner. Replacing the second output interval by $[0.90, 0.92]$ prevents that conclusion. If both scores are instead exactly 0.20, the first candidate wins the tie and only its output interval is needed. These are interval-information illustrations, not experimental instances.

3.1 DELETION-SPECIFIC PARAMETER AND VALIDATION REGIONS

Let $\widetilde{\theta}_h$ be any approximation to the retained-data minimizer and $r_h = \nabla R_h(\widetilde{\theta}_h; D^-)$. Strong convexity gives the classical suboptimal-model ball (Suzuki et al., 2014)

$$\theta_h(D^-) \in B(c_h, \rho_h), \qquad c_h = \widetilde{\theta}_h - \frac{r_h}{2\mu_h}, \qquad \rho_h = \frac{\|r_h\|}{2\mu_h}. \tag{7}$$

This established region is contained in the residual ball centered at $\widetilde{\theta}_h$ with radius $\|r_h\| / \mu_h$.

Suppose V_h is convex and β_h-smooth on the ball. Put $d_h = \nabla V_h(c_h)$. Then valid score bounds are

$$L_h = V_h(c_h) - \|d_h\| \rho_h, \qquad U_h = V_h(c_h) + \|d_h\| \rho_h + \frac{\beta_h}{2}\rho_h^2. \tag{8}$$

A known lower bound on V_h, such as zero, can tighten L_h. For a linear output $g_h(\theta) = t_h^\top \theta$, take $a_h = t_h^\top c_h - \|t_h\| \rho_h$ and $b_h = t_h^\top c_h + \|t_h\| \rho_h$. For a smooth nonlinear output, a Taylor remainder gives an analogous enclosure. Appendix C supplies the derivations, logistic constants, and a computable error bound for an influence warm start.

3.2 CONDITIONING THE PREDICTION REGION ON WINNING

In a learning problem, the score and output evaluate the same parameter, so their intervals need not vary independently. A winning candidate must satisfy $V_h(\theta_h(D^-)) \le u$. Convexity therefore implies

$$\theta_h(D^-) - c_h \in C_h(u) = \{e : \|e\| \le \rho_h,\ d_h^\top e \le u - V_h(c_h)\} \quad \text{whenever } h = h_*. \tag{9}$$

An empty $C_h(u)$ excludes h, but this is exactly the condition $L_h > u$ for the tangent lower bound in (8), not an additional pruning mechanism. For a nonempty region, bound the output by the extrema of $g_h(c_h + e)$ over that region. These bounds are *conditional on winning*. They may exclude a losing candidate's output without invalidating (5).

Lemma 3 (Dome support, adapted from Xiang et al. (2016))**.** *Let $C = \{e : \|e\| \le \rho,\ d^\top e \le b\}$ be nonempty with $d \neq 0$. Set $\alpha = b/\|d\|$, $t_\parallel = t^\top d/\|d\|$, and $t_\perp = (\|t\|^2 - t_\parallel^2)^{1/2}$. Its support in direction t is*

$$s_C(t) = \begin{cases} \rho\,\|t\|\,, & \rho t_\parallel \le \alpha\,\|t\|\,, \\ \alpha t_\parallel + \sqrt{\rho^2 - \alpha^2}\,t_\perp, & \textit{otherwise}. \end{cases} \tag{10}$$

Hence the winning candidate's output lies in $[t^\top c - s_C(-t),\ t^\top c + s_C(t)]$.

The second case has an active halfspace boundary and $-\rho \le \alpha < \rho$. For $d = 0$, the region is the full ball when $b \ge 0$ and empty otherwise. The result includes $t = 0$ and $\rho = 0$ with the corresponding zero support. The formula is Lemma 3, Eq. (27), of Xiang et al. (2016), with the halfspace written in our coordinates. Appendix D includes the derivation and degenerate cases for completeness. The additional work is $O(p_h)$ per queried linear output once the validation gradient is available.

4 Adaptive refinement and its computational limits

The full-data model bank and its Hessian factorizations are shared across deletion queries. At candidate h, a normalization-aware influence warm start is

$$\widetilde{\theta}_h^{(0)} = \theta_h(D) - H_h(D)^{-1}\nabla R_h(\theta_h(D); D^-), \qquad H_h(D) = \nabla^2 R_h(\theta_h(D); D). \tag{11}$$

The certificate uses the actual retained-data residual, so it does not assume that the influence expansion is accurate. Any warm start may be substituted. The logistic implementation uses damped Newton, but the certificate accepts any routine that produces new approximations. Equation (11) requires twice differentiability; Appendix G gives the additional local rate conditions. A high-accuracy numerical fit does not supply an exact score comparison.

Algorithm 1 Switch-aware attribution (SWA)

Require: Ordered candidate menu, deletion G, output tolerance ε, baseline interval
1: Initialize all candidates by (11) and set $\mathcal{S} \leftarrow \mathcal{H}$
2: **for** $k = 0, 1, \ldots$ up to the computational budget **do**
3: Compute residual, score, and output bounds for every $h \in \mathcal{S}$
4: Intersect unconditional bounds and check their consistency
5: $u \leftarrow \min_{h \in \mathcal{S}} U_h$
6: Keep the menu-aware feasible winners in (6), restricted to $\mathcal{S}$
7: Accumulate winner-conditional output bounds from (9)–(10)
8: Remove candidates with empty conditional output intersections
9: Form the attribution enclosure I_G by (5)
10: **if** every coordinate of I_G has width at most 2ε **then**
11: **return** midpoint of I_G and its enclosure
12: **end if**
13: Refine each $h \in \mathcal{S}$ once using a solver satisfying its declared assumptions
14: **end for**
15: **return** the enclosure and an unresolved flag, without a point estimate

Theorem 4 (Safety under adaptive refinement)**.** *If every candidate bound and baseline interval is valid, Algorithm 1 never removes the true selected candidate and every returned enclosure contains Δ_G. Every returned point estimate has absolute error at most ε, or ℓ_∞ error at most ε for a vector of outputs. These conclusions hold for arbitrary valid refinement schedules and do not require a positive selection gap.*

The invariant proof is in Appendix E. Score intervals contain each candidate's true score. Output intervals may instead be conditional on winning, so their accumulated intersection must contain

the output only of the actual winner. Eliminated candidates cannot later win because refinement changes information about a fixed counterfactual, not the counterfactual itself. Exact ties use the strict comparison against earlier menu indices in (6). Numerical near-equality is not treated as equality. An empty unconditional intersection signals invalid evidence. An empty winner-conditional intersection instead excludes the candidate. Lemma 15 formalizes this distinction, with a smooth regression example. An unresolved query returns no point estimate.

4.1 Output disagreement, not only the selection gap

Write $v_* = \min_h v_h$ and define the near-optimal output spread

$$\omega(t) = \max_{h: v_h - v_* \le t} f_h - \min_{h: v_h - v_* \le t} f_h, \qquad t \ge 0. \tag{12}$$

This is a task-specific measure of predictive multiplicity (Marx et al., 2020). It enters the computational guarantee directly.

Theorem 5 (Output-dependent refinement bound)**.** *Use the synchronous schedule in Algorithm 1, refining every active candidate exactly once per round. Assume that a candidate refined k times has an unconditional score interval of width at most Bq^k and an unconditional output interval of width at most Aq^k, where $A, B > 0$ and $0 < q < 1$. Let $d_0 = \overline{F}_0 - \underline{F}_0$. After round k, Algorithm 1 has*

$$\operatorname{diam}(I_G^{(k)}) \le \omega(2Bq^k) + 2Aq^k + d_0. \tag{13}$$

For any integer T with right-hand side at most 2ε, it stops by round T. Put $\tau_h = \min\{k \ge 0 : 2Bq^k < v_h - v_*\}$, with $\tau_h = \infty$ if the set is empty. The number of candidate refinement steps is at most

$$\sum_{h=1}^{K} \min\{T, \tau_h\}. \tag{14}$$

A survivor obeys $v_h \le L_h + Bq^k \le U_{h_*} + Bq^k \le v_* + 2Bq^k$. Its true output therefore contributes to $\omega(2Bq^k)$, and its two endpoints add at most $2Aq^k$. Candidates with $2Bq^k < v_h - v_*$ have already been eliminated. Appendix E gives the full argument and the coordinatewise extension.

The geometric-width assumption is needed for this work bound, not for safety. It does not imply a common global contraction factor for all damped Newton trajectories. With a unique winner, sufficiently small score widths eventually isolate that candidate. With a tie, finite identification may fail, while attribution can still terminate when tied candidates agree sufficiently closely on the output. If plausible winners disagree by more than 2ε, the minimax statement in Theorem 2 shows why interval information alone cannot justify a narrower point estimate.

4.2 A separation generated by a finite training sample

The interval-information distinction can arise from the residual regions of an actual ridge problem, without prescribing score and output intervals independently.

Proposition 6 (Finite-data separation for a specified refinement rule)**.** *Take the three training pairs*

$$(\sqrt{2}e_1, 0), \qquad (\sqrt{2}e_2, \sqrt{2}), \qquad (e_1, 1),$$

delete the third, and use the normalized squared-loss objective (1) *with ordered menu* $(1, 3)$*. Let validation consist of $(e_2, 3/8 + \delta)$, where $0 < \delta < 1/200$, and let $g(\theta) = e_1^\top \theta$. Initialize by* (11)*, use the residual regions with $\mu_h = \lambda_h$, and halve the exact Newton displacement at each synchronous round. Then $\Delta_G = -1/6$. Rectangular output stopping needs at most*

$$2 \max\left\{0, \left\lceil \log_2 \frac{1}{50\varepsilon} \right\rceil\right\}$$

candidate refinements. With the same initialization, score bounds and schedule, a race requiring a winner needs at least $2\lceil \log_2(1/(200\delta)) \rceil$ refinements. In contrast, one full Newton step per candidate resolves both tasks.

Appendix I computes all fits, residuals, and bounds. The gap is $\delta/4$, whereas both retained predictions are zero. The separation applies to the stated residual-based races, not to arbitrary algorithms with data access. A direct normal-equation solve is competitive on this small problem. The example isolates the interaction between output agreement and refinement granularity; it does not recommend damping a quadratic solve to obtain an advantage.

5 EXPERIMENTS

We evaluate three questions. Does reselection change attribution? When does output stopping save refinements over equally certified winner identification? Do fewer refinements reduce complete query time? The main and dense-grid results use the implementation of Algorithm 1. The research companion records the splits, deletion groups, candidate scores, output bounds, and termination histories.

Data, models, and comparators. The main study uses breast cancer, wine, and digits from scikit-learn (Pedregosa et al., 2011), plus binary logistic heads on three frozen digits checkpoints trained on a disjoint upstream subset. Raw-feature menus have 19 regularizers in $[10^{-6}, 1]$. The checkpoint menu has 13 per representation, or 39 candidates. Ten stratified splits per setting use approximately 60/20/20% of the head data. Each split has 20 deletions for every combination of $1\%, 5\%, 10\%$ and uniform or neighborhood removal. The resulting 1,200 queries per setting each attribute 16 held-out signed logits. Features, preprocessing, menu, validation data, and training randomness remain fixed within a counterfactual.

Fixed IF and *Fixed exact* retain the original configuration. *Reselect IF*, *Reselect Newton*, and *IF + Newton* approximate every candidate and reselect by its evaluated validation loss. The last two use one retained-objective Newton step from the original fit or the influence start, respectively. The numerical reference refits all candidates to $\|\nabla R_h\| / \lambda_h \leq 10^{-9}$. Score intervals separate every main and dense reference winner. The main certified comparator uses the same warm starts, score pruning, schedule, and output tolerance as SWA, but requires one surviving candidate. A rectangular variant omits conditioning. A uniform variant continues refining every candidate. Appendix J specifies all details, including the restricted TRAK-style ranking diagnostic.

Table 1: Main joint-logit study, with 1,200 queries per setting. Percentages have descriptive 95% Student-t half-widths across ten seeded splits. Failure means maximum attribution error above 10^{-3}. Maxima are rounded upward.

Setting	K	Reselection (%)	IF + Newton failure (%)	Max. SWA error
Breast cancer	19	19.1 ± 9.1	40.3 ± 9.3	6.66×10^{-4}
Wine	19	13.0 ± 11.4	45.0 ± 4.3	7.68×10^{-4}
Digits	19	17.2 ± 7.8	49.0 ± 5.9	6.68×10^{-4}
Frozen checkpoints	39	12.3 ± 9.4	45.5 ± 6.9	8.34×10^{-4}

Reselection and predictive quality. The winner changes in 12.3–19.1% of queries by setting (Table 1). Exact refitting of the old winner has only 55.7–64.8% mean sign accuracy on switched queries, despite answering stable queries correctly. Thus reselection is distinct from approximation error. Selected test accuracy before deletion ranges from 89.4% to 98.6% across the four binary tasks. Appendix M reports post-deletion losses, score margins, selected regularizers, and the separate within-configuration and reselection effects. Wine selects a grid endpoint on 69.2% of retained queries, limiting the interpretation of these switches as well-resolved interior tuning.

Stringent joint accuracy leaves no additional saving. SWA meets the 10^{-3} point-error tolerance on all 4,800 queries. Conditional, rectangular, and winner-required races have identical refinement counts and finish with one survivor. Against uniform refinement in the 600-query allocation control, the saving is 63.0–72.9% and comes from *candidate elimination*, not conditioning or output stopping. In 360 dense-menu evaluations, representing 60 deletions reused across two menus and three tolerances, output stopping saves only six of 58,086 winner-required refinements. Conditioning saves none. These controls attribute the saving to elimination rather than to the output stopping rule.

Probability tolerance and the number of outputs. We use breast cancer and digits, three splits each, 12 fixed deletion groups spanning all six size/mechanism conditions, menus of 19 or 65 candidates, and regularization ranges $[10^{-6}, 1]$ or $[10^{-3}, 1]$. No deletion is selected by validation margin, output alignment, or observed savings. The target is the change in *true-label probability*, obtained by transforming pre- and post-deletion signed-logit intervals separately. We evaluate

$\varepsilon \in \{10^{-4}, 10^{-3}, 10^{-2}, 0.05\}$, both individually for every one of the 16 held-out targets and jointly for all 16. Individual results cover 4,608 case/menu/target combinations per tolerance, not 4,608 independent deletions.

Table 2: Probability-output work frontier. Scalar and joint savings compare conditional stopping with the equally certified winner race. "Cap increment" compares conditional with rectangular stopping. Cases helped count strict scalar work reductions. Sign reports exclude zero from the returned interval. Every zero-gain case is included.

ε	Scalar saving	Cases helped	Cap increment	Joint saving	Sign reports
0.0001	1.49%	212/4608	0.09%	0.00%	94.7%
0.001	2.99%	396/4608	0.40%	0.00%	89.6%
0.01	7.00%	881/4608	0.73%	0.33%	75.8%
0.05	12.36%	1487/4608	1.50%	2.60%	60.0%

Table 2 reports 1.49–12.36% fewer scalar refinements, although most cases save no steps. Requiring all outputs simultaneously reduces the saving to 0–2.60%. At $\varepsilon = 0.01$, individual targets save 7.00% of refinements and permit sign reports for 75.8% of targets; the joint saving is 0.33%. Conditioning adds only 0.73% relative to rectangular stopping at that tolerance. Appendix M reports split-level gains, and the research records include every menu/tolerance combination. At 0.05, scalar gains range from 2.10% to 26.29% across the six splits. The pooled ratio therefore conceals substantial variation.

Decisions, approximation, and the mechanism. For each target we also compare every pair among the four same-fraction deletion groups. At $\varepsilon = 0.01$, nonoverlapping intervals determine 65.4% of these orders, with no incorrect assertion against the numerical reference. Tighter tolerance increases reporting but costs more work. A point estimate that reselects among the same active candidates after the same number of refinements also meets all tested probability tolerances. It often has smaller median error than the interval midpoint. The certificate justifies the requested accuracy; it does not improve observed point-estimation error in this comparison. Appendix M reports complete trajectories, pre-refit resolution rates, and prediction disagreement and cap activity beside validation gaps.

The finite-data experiment varies δ, output alignment η, tolerance, and damping independently. With half Newton steps, $\delta = 10^{-8}$, aligned output, and $\varepsilon = 10^{-3}$, output certification takes 10 refinements versus 42 for winner identification. Increasing output disagreement removes this advantage. With full Newton steps, every method takes two refinements. This control ties Proposition 6 to the actual certificate while exposing why a fast solver can erase the separation.

Complete costs and larger sample sizes. A separate study uses 8,640 complete-query timings with four counterbalanced method orders. Retained-data construction, initialization, candidate construction, certificate evaluation, and refinement are all timed. On individually queried probabilities, rectangular stopping takes 95.1–98.1% of the winner race's pooled time across the three tolerances; conditioning takes 105.3–109.2%. The small additional refinement saving does not offset the cost of conditioning. Model-bank time and storage are reported separately. A controlled 65-parameter logistic simulation increases training size from 2,000 to 20,000, with 17 candidates and source-correlated label noise. It meets the requested accuracy and shows substantial elimination savings, but negligible additional output-stopping benefit at 20,000 records. All six simulated banks select the weakest regularizer, so the study provides a limited test of difficult model reselection. It is not an external-data benchmark.

Numerical and statistical interpretation. Theorems concern valid bounds in real arithmetic. The implementation uses float64 guards, not directed rounding. Twenty main reference centers lie slightly outside the reported intervals, by at most 1.06×10^{-10}; every interval overlaps its reference uncertainty interval and every point estimate meets the requested tolerance. Audits use a declared 10^{-8} allowance. Overlapping deletions, repeated menus, and shared targets are not independent replicates. These comparisons describe the stated protocols. They do not establish population coverage or a general runtime theorem.

6 Related work and scope

Influence-based attribution and the infinitesimal jackknife approximate how fitted parameters respond to changes in observation weights (Koh & Liang, 2017; Giordano et al., 2019). Group-deletion error (Koh et al., 2019) and discrepancies between influence targets and literal leave-one-out retraining (Bae et al., 2022) motivate specifying the counterfactual carefully. Data Shapley and datamodels permit a general learning procedure (Ghorbani & Zou, 2019; Ilyas et al., 2022). Our target is one specified deletion, rather than a Shapley allocation over all coalitions.

Influence-based model selection (Debruyne et al., 2008), guaranteed approximate cross-validation (Wilson et al., 2020), validation bounds from suboptimal models (Suzuki et al., 2014), and non-stochastic hyperparameter racing (Jamieson & Talwalkar, 2016) provide the computational foundations. Suzuki et al. derive the shifted parameter ball, prediction bounds, and validation-based candidate exclusion. Wilson et al. give assessment and model-selection guarantees for approximate cross-validation; our target is the selected output after one stipulated deletion. The ball–halfspace support formula is also established (Xiang et al., 2016, Lemma 3). These bounds, their geometry, and candidate elimination are inherited ingredients. Our results use them to characterize selected-output partial identification, condition output regions on winning, and bound refinements through near-optimal prediction disagreement. This disagreement is related to predictive multiplicity (Marx et al., 2020), but here determines a deletion-attribution stopping rule.

Wang et al. (2025) study the sensitivity of attribution estimators to their own hyperparameters. Our candidate menu instead belongs to the predictive training procedure and is reselected after deletion. TRAK uses projected model features for scalable attribution (Park et al., 2023). Influence attribution for diffusion models (Mlodozeniec et al., 2025) and stagewise attribution during learning (Lee et al., 2026) concern other model classes and training dynamics. The guarantees here require strong convexity and do not cover arbitrary neural optimization.

The individual-probability study identifies a qualified benefit of output stopping. The target and tolerance determine which prediction differences matter, while the optimizer determines whether tighter bounds avoid a refinement. The finite-data construction isolates both effects. On the classification grids, output stopping saves modest work for individual targets and much less for joint targets. Geometric conditioning frequently tightens endpoints but saves few refinements and increases measured latency. Point approximations evaluated at the same refinement counts are already accurate. The certificate's demonstrated value is a justification for termination, rather than a better point estimate.

The experiments cover small public classification problems, convex heads on fixed features, and a larger controlled simulation. The simulation selects a regularization boundary and does not establish external-data effectiveness. Seed variation measures repeated splits of fixed datasets, not sampling from a new population. The guarantees require a specified menu and valid numerical regions, and the implementation is not outward-rounded. Within these limits, selected-output attribution is well defined and admits a certificate. Whether the certificate saves computation depends jointly on output agreement, requested precision, and refinement granularity. A more elaborate certifying procedure is useful only when these savings justify its additional cost.

AI USE STATEMENT

Generative AI was used for proofreading, including grammatical correction, text enhancement, and general “polishing". The authors assume full responsibility for the correctness and originality of the research.

ETHICS STATEMENT

This work studies counterfactual changes within a specified learning procedure. An attribution certificate is not a certificate of privacy, legal entitlement, data ownership, or fairness. The experiments use public classification datasets and do not establish clinical utility. The neural-feature experiments do not remove information from the upstream feature extractor. Interpreting the method as end-to-end machine unlearning would therefore be incorrect.

RESEARCH ARTIFACTS

The accompanying research archive contains the implementation, original and current experiment records, split and deletion identifiers, solver histories, verification programs, and table-generation scripts. Appendix J specifies the main protocol, and Appendix M specifies the additional studies and unified execution path. Complete proofs include the conditional-infeasibility and finite-data separation arguments in Appendices H and I. The original archive is retained for provenance. The polished manuscript uses the same results as the current result directory in the research companion.

REFERENCES

Juhan Bae, Nathan Ng, Alston Lo, Marzyeh Ghassemi, and Roger Grosse. If influence functions are the answer, then what is the question? In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://arxiv.org/abs/2209.05364.

Michiel Debruyne, Mia Hubert, and Johan A. K. Suykens. Model selection in kernel based regression using the influence function. *Journal of Machine Learning Research*, 9(78):2377–2400, 2008. URL https://jmlr.org/papers/v9/debruyne08a.html.

Amirata Ghorbani and James Zou. Data Shapley: Equitable valuation of data for machine learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 2242–2251. PMLR, 2019. URL https://proceedings.mlr.press/v97/ghorbani19c.html.

Ryan Giordano, William Stephenson, Runjing Liu, Michael Jordan, and Tamara Broderick. A Swiss Army infinitesimal jackknife. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, volume 89, pp. 1139–1147. PMLR, 2019. URL https://proceedings.mlr.press/v89/giordano19a.html.

Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Understanding predictions with data and data with predictions. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 9525–9587. PMLR, 2022. URL https://proceedings.mlr.press/v162/ilyas22a.html.

Kevin Jamieson and Ameet Talwalkar. Non-stochastic best arm identification and hyperparameter optimization. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51, pp. 240–248. PMLR, 2016. URL https://proceedings.mlr.press/v51/jamieson16.html.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 1885–1894. PMLR, 2017. URL https://proceedings.mlr.press/v70/koh17a.html.

Pang Wei Koh, Kai-Siang Ang, Hubert H. K. Teo, and Percy Liang. On the accuracy of influence functions for measuring group effects. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://arxiv.org/abs/1905.13289.

Jin Hwa Lee, Matthew Smith, Maxwell Adam, and Jesse Hoogland. Influence dynamics and stagewise data attribution. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/3c1edb917edb6cec59bf3a0afc03150e-Abstract-Conference.html.

Charles Marx, Flavio Calmon, and Berk Ustun. Predictive multiplicity in classification. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 6765–6774. PMLR, 2020. URL https://proceedings.mlr.press/v119/marx20a.html.

Bruno Mlodozeniec, Runa Eschenhagen, Juhan Bae, Alexander Immer, David Krueger, and Richard E. Turner. Influence functions for scalable data attribution in diffusion models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/804dbf8d3b8eee1ef875c6857efc64eb-Abstract-Conference.html.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. TRAK: Attributing model behavior at scale. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 27074–27113. PMLR, 2023. URL https://proceedings.mlr.press/v202/park23c.html.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. URL https://jmlr.org/papers/v12/pedregosa11a.html.

Yoshiki Suzuki, Kohei Ogawa, Yuki Shinmura, and Ichiro Takeuchi. An algorithmic framework for computing validation performance bounds by using suboptimal models. arXiv preprint arXiv:1402.2148, 2014. URL https://arxiv.org/abs/1402.2148.

Weiyi Wang, Junwei Deng, Yuzheng Hu, Shiyuan Zhang, Xirui Jiang, Runting Zhang, Han Zhao, and Jiaqi Ma. Taming hyperparameter sensitivity in data attribution: Practical selection without costly retraining. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/f7dee2af577ab06037891f264157d718-Abstract-Conference.html.

Ashia Wilson, Maximilian Kasy, and Lester Mackey. Approximate cross-validation: Guarantees for model assessment and selection. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, volume 108, pp. 4530–4540. PMLR, 2020. URL https://proceedings.mlr.press/v108/wilson20a.html.

Zhen James Xiang, Yun Wang, and Peter J. Ramadge. Screening tests for lasso problems. arXiv:1405.4897v2, 2016. URL https://arxiv.org/abs/1405.4897v2. Version 2, 21 August 2016; first posted in 2014.

A COUNTERFACTUAL SPECIFICATION AND A BOUNDED RIDGE EXAMPLE

A.1 OBJECTS HELD FIXED BY AN INTERVENTION

A deletion query is evaluated conditionally on a finite, ordered candidate menu and a fixed validation sample. Candidate-specific feature maps may have different dimensions. Feature maps, fitted transformations, upstream training samples, and random seeds are part of the conditioning information. The retained objective is then exactly the function in (1), with denominator $n - s$. The same regularization coefficient appears before and after deletion. This convention differs from holding fixed the coefficient of a *summed* training loss.

Every R_h is differentiable and strongly convex on all of $\mathbb{R}^{p_h}$. Strong convexity implies coercivity through

$$R_h(\theta) \geq R_h(0) + \nabla R_h(0)^\top \theta + \frac{\mu_h}{2} \|\theta\|^2 .$$

Continuity and coercivity give existence of a minimizer, and strong convexity gives uniqueness. The finite menu and deterministic tie rule make the selected candidate well defined. No randomness remains in the mathematical counterfactual after the conditioning information is fixed.

The attributed quantity may be a prediction, a loss, or another scalar evaluation, provided the corresponding output enclosure is valid. The selected validation score itself and the prediction on a separate example are not interchangeable outputs. In particular, two candidates can have indistinguishable validation scores and different attributed predictions.

A.2 PROOF OF PROPOSITION 1

Proof. Fix $n \geq 3$ and set every training covariate to 1. The first $n - 1$ responses equal

$$y_i = 1 - \frac{1}{2(n-1)},$$

and the final response is $y_n = 2$. Delete $G = \{n\}$. Every response belongs to $[3/4, 2]$, uniformly in n. For squared loss, the full and retained empirical response means are

$$m = \frac{(n-1)(1 - 1/[2(n-1)]) + 2}{n} = 1 + \frac{1}{2n}, \qquad m^- = 1 - \frac{1}{2(n-1)}.$$

The regularized mean objective has derivative $(1 + \lambda)\theta - m$, hence its minimizer is $m/(1 + \lambda)$. The two candidates therefore predict $m/2$ and $m/4$ for $\lambda = 1$ and $\lambda = 3$.

Take one validation observation with covariate 1, response $3/8$, and loss $V(\theta) = (\theta - 3/8)^2/2$. Direct expansion gives

$$\begin{aligned} V(m/2) - V(m/4) &= \frac{1}{2}\left[(m/2 - 3/8)^2 - (m/4 - 3/8)^2\right] \\ &= \frac{1}{2}\left(\frac{m}{4}\right)\left(\frac{3m}{4} - \frac{3}{4}\right) = \frac{3m(m-1)}{32}. \end{aligned}$$

Since $m > 1$, the original selected candidate is $\lambda = 3$. Since $0 < m^- < 1$, the retained selected candidate is $\lambda = 1$. Both selections are strict. Taking $g(\theta) = \theta$, the selected effect equals

$$\frac{m^-}{2} - \frac{m}{4} = \frac{1}{4} - \frac{1}{4(n-1)} - \frac{1}{8n}.$$

It is at least its value at $n = 3$, namely $1/12$, since both subtracted terms decrease with n. It tends to $1/4$. Keeping the original selected candidate instead gives

$$\frac{m^-}{4} - \frac{m}{4} = -\frac{1}{8(n-1)} - \frac{1}{8n},$$

which is strictly negative and tends to zero. The deletion fraction is $1/n$. All training losses are strongly convex after regularization, all observations are bounded, and the validation observation and regularization menu are independent of n. □

The construction approaches a validation tie as n increases. Accordingly, it does not contradict a guarantee that assumes a fixed positive validation gap. It shows that bounded data, strong convexity, and a small deletion fraction alone do not control the selected-output effect. The decomposition (3) follows by adding and subtracting $g_{h_0}(\theta_{h_0}(D^-))$ and does not require convexity.

B SELECTED-OUTPUT SETS UNDER INTERVAL INFORMATION

B.1 CONTAINMENT, DETERMINISTIC TIES, AND SHARPNESS

Proof of Theorem 2. Let h_* be the actual selected candidate. For each j, interval validity and optimality give

$$L_{h_*} \le v_{h_*} \le v_j \le U_j.$$

Thus $L_{h_*} \le \min_j U_j$ and $h_* \in \mathcal{S}$. Since $f_{h_*} \in [a_{h_*}, b_{h_*}]$, its value is between $\min_{h\in\mathcal{S}} a_h$ and $\max_{h\in\mathcal{S}} b_h$. Subtracting any baseline in $[\underline{F}_0, \overline{F}_0]$ proves (5). This argument also holds if each output interval is only guaranteed to contain f_h on the event that $h = h_*$.

For the exact feasible-winner characterization, suppose first that candidate h wins for some vector $(v_1, \dots, v_K)$ in the product of score intervals. For every j, $L_h \le v_h \le v_j \le U_j$. When $j < h$, a tie would favor j, so $v_h < v_j$ and consequently $L_h < U_j$. This proves necessity in (6).

Conversely, suppose that h satisfies (6). Set $v_h = L_h$ and $v_j = U_j$ for $j \ne h$. The resulting score vector belongs to the product of intervals. Candidate h has score no larger than any competitor and has strictly smaller score than any lower-index competitor. It is therefore selected under the specified tie rule. This proves sufficiency, including degenerate point intervals.

Now impose the rectangular-information assumption that all score coordinates, all output coordinates, and the baseline can be chosen independently within their respective intervals. For every feasible winner h, its score vector can be chosen as above independently of f_h and F_0. The set of differences of two closed intervals is

$$[a_h, b_h] - [\underline{F}_0, \overline{F}_0] = [a_h - \overline{F}_0, b_h - \underline{F}_0].$$

This interval is the image of a connected rectangle under the continuous linear map $(x, y) \mapsto x - y$. The stated minimum and maximum are attained, so every intermediate value is achievable. Taking the union over exactly the feasible winners gives the asserted set, and no other values are possible.

Write A and B for the smallest and largest points of this nonempty compact union. For any proposed scalar estimate z, the worst-case error is at least

$$\max\{|z - A|, |z - B|\} \ge \frac{B - A}{2},$$

where the inequality follows from $|z - A| + |z - B| \ge B - A$. The midpoint $z = (A + B)/2$ is within $(B - A)/2$ of every point of $[A, B]$, and hence of every feasible attribution. It attains the lower bound. Gaps within the union do not alter this absolute-error minimax radius. □

Independence is used only for sharpness, not containment. Training data typically couple validation scores to predictions, and different candidates may also share parameters or sufficient statistics. The rectangular model discards those relations. A smaller interval obtained from additional valid structure does not contradict the sharpness result.

B.2 VECTOR OUTPUTS, SIGNS, AND COMPARISONS

For q outputs, let a_{hr} and b_{hr} be valid coordinatewise bounds for candidate h and coordinate r. Use the same feasible-winner superset for every coordinate and define

$$A_r = \min_{h\in\mathcal{S}} a_{hr} - \overline{F}_{0r}, \qquad B_r = \max_{h\in\mathcal{S}} b_{hr} - \underline{F}_{0r}.$$

The actual vector belongs to $\prod_{r=1}^{q} [A_r, B_r]$. Its coordinatewise midpoint has ℓ_∞ error at most $\frac{1}{2}\max_r (B_r - A_r)$. The box can contain vectors that no single winning candidate realizes. This does not invalidate containment or the stated error bound. For any compact vector-valued feasible set,

every extreme coordinate is attained somewhere in the set. The bounding-box midpoint therefore achieves the minimum worst-case ℓ_∞ radius, even when the midpoint itself is infeasible.

If a scalar attribution interval has lower endpoint strictly positive, every feasible effect is positive. If its upper endpoint is strictly negative, every feasible effect is negative. When it contains zero, a midpoint sign is not certified. Similarly, two scalar interventions with intervals $[A_1, B_1]$ and $[A_2, B_2]$ have certified ordering $\Delta_1 < \Delta_2$ whenever $B_1 < A_2$. Dependence between the interventions does not affect this sufficient condition.

C Residual regions and deletion-aware warm starts

C.1 Strong-convexity and validation regions

The shifted parameter ball below is an established suboptimal-model region, used here with the retained objective (Suzuki et al., 2014). We include the proof to specify exactly which assumptions the certificate uses.

Lemma 7 (Shifted residual ball). *Let R be differentiable and μ-strongly convex on $\mathbb{R}^p$, and let θ^* minimize it. For an arbitrary $\widetilde{\theta}$, put $r = \nabla R(\widetilde{\theta})$. Then*

$$\left\|\theta^* - \widetilde{\theta} + r/(2\mu)\right\| \le \frac{\|r\|}{2\mu}.$$

In particular, $\left\|\theta^* - \widetilde{\theta}\right\| \le \|r\|/\mu$.

Proof. The two strong-convexity inequalities at x and y, added together, yield strong monotonicity

$$(\nabla R(x) - \nabla R(y))^\top (x - y) \ge \mu \|x - y\|^2.$$

Take $x = \widetilde{\theta}$, $y = \theta^*$, and use $\nabla R(\theta^*) = 0$. With $e = \theta^* - \widetilde{\theta}$, the result is $r^\top e \le -\mu \|e\|^2$. Completing the square gives

$$\|e + r/(2\mu)\|^2 = \|e\|^2 + \frac{r^\top e}{\mu} + \frac{\|r\|^2}{4\mu^2} \le \frac{\|r\|^2}{4\mu^2}.$$

The final bound follows either from the triangle inequality or from strong monotonicity and Cauchy–Schwarz. The shifted ball is contained in the ordinary ball centered at $\widetilde{\theta}$ with radius $\|r\|/\mu$, because its center is displaced by half that radius and its own radius is the other half. □

Lemma 8 (Propagation to validation and outputs). *Suppose $\theta^* \in B(c, \rho)$. If V is differentiable, convex, and has a β-Lipschitz gradient on the ball, then*

$$V(c) - \|\nabla V(c)\|\rho \le V(\theta^*) \le V(c) + \|\nabla V(c)\|\rho + \frac{\beta}{2}\rho^2.$$

For a linear output $t^\top\theta$, the extremal values on the ball are $t^\top c \pm \|t\|\rho$. More generally, if g has an M_g-Lipschitz gradient on the ball, then

$$|g(\theta^*) - g(c)| \le \|\nabla g(c)\|\rho + \frac{M_g}{2}\rho^2.$$

Proof. Write $e = \theta^* - c$ with $\|e\| \le \rho$. Convexity gives $V(c+e) \ge V(c) + \nabla V(c)^\top e$. The smoothness upper bound is

$$V(c+e) \le V(c) + \nabla V(c)^\top e + \frac{\beta}{2}\|e\|^2.$$

Apply Cauchy–Schwarz to the linear term in both inequalities. The extrema for a linear output follow from Cauchy–Schwarz, with equality at $e = \pm\rho t/\|t\|$ when $t \ne 0$. For the nonlinear output, the integral remainder satisfies

$$\begin{aligned}|g(c+e) - g(c) - \nabla g(c)^\top e| &\le \int_0^1 \|\nabla g(c+te) - \nabla g(c)\|\,\|e\|\,dt \\ &\le \int_0^1 M_g t\|e\|^2\,dt = \frac{M_g}{2}\|e\|^2.\end{aligned}$$

No convexity of g is required for this last claim. □

C.2 NORMALIZATION-AWARE DELETION RESIDUAL

Let θ_0 be the exact full-data minimizer of $R(\theta; D) = n^{-1}\sum_{i\in D}\ell_i(\theta) + \lambda\|\theta\|^2/2$. Define $g_i = \nabla\ell_i(\theta_0)$. Full-data stationarity gives $\sum_{i\in D} g_i = -n\lambda\theta_0$. Thus

$$\begin{aligned} r_0^- = \nabla R(\theta_0; D^-) &= \frac{-n\lambda\theta_0 - \sum_{i\in G} g_i}{n-s} + \lambda\theta_0 \\ &= -\frac{\sum_{i\in G} g_i + s\lambda\theta_0}{n-s}. \end{aligned} \tag{15}$$

Dropping $s\lambda\theta_0$ or replacing $n-s$ by n changes the objective being approximated. Exact original stationarity is not necessary. For any approximate original fit $\bar\theta$, put $e_D = \nabla R(\bar\theta; D)$. Substituting $\sum_{i\in D}\nabla\ell_i(\bar\theta) = n(e_D - \lambda\bar\theta)$ gives the exact identity

$$\nabla R(\bar\theta; D^-) = \frac{ne_D - \sum_{i\in G}\nabla\ell_i(\bar\theta) - s\lambda\bar\theta}{n-s}. \tag{16}$$

The original optimization residual is amplified by $n/(n-s)$ and cannot simply be discarded. The implementation evaluates the retained gradient directly rather than relying on exact full-data stationarity.

When the losses are twice differentiable and $s > 0$, write $H_D^{\mathrm{data}} = n^{-1}\sum_{i\in D}\nabla^2\ell_i(\theta_0)$ and $H_G^{\mathrm{data}} = s^{-1}\sum_{i\in G}\nabla^2\ell_i(\theta_0)$. Then

$$H^-(\theta_0) - H(\theta_0; D) = \frac{s}{n-s}\left(H_D^{\mathrm{data}} - H_G^{\mathrm{data}}\right). \tag{17}$$

Indeed, subtracting H_D^{data} from $(nH_D^{\mathrm{data}} - sH_G^{\mathrm{data}})/(n-s)$ yields the stated expression, and the regularization Hessian cancels.

Lemma 9 (A posteriori influence warm-start bound)**.** *Let H_0 be an invertible matrix, let $r_0^- = \nabla R^-(\theta_0)$, and choose a displacement e. Set $\xi = r_0^- + H_0 e$. Suppose the retained Hessian is M-Lipschitz in operator norm on the segment from θ_0 to $\theta_0 + e$. Then*

$$\left\|\nabla R^-(\theta_0 + e)\right\| \le \|\xi\| + \left\|H^-(\theta_0) - H_0\right\|_2\|e\| + \frac{M}{2}\|e\|^2. \tag{18}$$

If R^- is μ-strongly convex, dividing the right-hand side by μ bounds the parameter error at $\theta_0 + e$.

Proof. The fundamental theorem of calculus gives

$$\begin{aligned} \nabla R^-(\theta_0 + e) &= r_0^- + H^-(\theta_0)e + \int_0^1 \big(H^-(\theta_0 + te) - H^-(\theta_0)\big)e\,dt \\ &= \xi + \big(H^-(\theta_0) - H_0\big)e + \int_0^1 \big(H^-(\theta_0 + te) - H^-(\theta_0)\big)e\,dt. \end{aligned}$$

The integral norm is at most $\int_0^1 Mt\|e\|^2\,dt = M\|e\|^2/2$. Apply the triangle inequality and Lemma 7. □

For an exact influence solve, $e = -H_0^{-1}r_0^-$ and $\xi = 0$. For an inexact linear solve, ξ is an observable residual rather than an unmodeled error. This lemma explains why an influence approximation can be an inexpensive starting point, but the algorithm evaluates the actual retained gradient at that point. Its certificate therefore does not require a small deletion, a correct first-order expansion, or an available Hessian-Lipschitz constant.

C.3 LOGISTIC CONSTANTS AND GENERALIZED LINEAR MODELS

For binary logistic loss with $y \in \{0, 1\}$,

$$\ell(y, x^\top\theta) = \log(1 + e^{x^\top\theta}) - yx^\top\theta, \qquad \sigma(t) = \frac{1}{1+e^{-t}}.$$

The Hessian of the mean regularized objective is

$$\nabla^2 R(\theta) = \frac{1}{m} X^\top \operatorname{diag}\{p_i(1-p_i)\} X + \lambda I, \qquad p_i = \sigma(x_i^\top \theta).$$

Since $0 \le p_i(1-p_i) \le 1/4$, the objective is λ-strongly convex. For an unregularized validation loss on design Z with m_v rows, a global smoothness constant is

$$\beta = \frac{\left\| Z^\top Z \right\|_2}{4m_v}. \tag{19}$$

These bounds include the intercept because the code appends it as a feature and penalizes it. An unpenalized intercept would require a different certified strong-convexity lower bound.

For completeness, the absolute third derivative of the scalar logistic loss is bounded by $1/(6\sqrt{3})$. To see this, write $p = \sigma(t)$ and consider $|p(1-p)(1-2p)|$ on $[0,1]$. By symmetry it suffices to maximize $p - 3p^2 + 2p^3$ on $[0, 1/2]$. Its derivative vanishes at $p = (3-\sqrt{3})/6$, which gives the stated maximum. Therefore a valid global Hessian-Lipschitz constant is

$$M \le \frac{1}{6\sqrt{3}\, m} \sum_{i=1}^{m} \|x_i\|^3 .$$

Indeed, each Hessian coefficient changes by at most $\|x_i\| \, \|\theta - \theta'\| / (6\sqrt{3})$, and $\left\| x_i x_i^\top \right\|_2 = \|x_i\|^2$. Summing proves the bound.

The residual argument also applies to other differentiable strongly convex generalized linear models. The validation smoothness and output bounds must hold throughout the region used in the calculation. Global constants are convenient but unnecessary. A local constant is valid only when its parameter region has also been established.

D Geometry of selection-conditioned predictions

D.1 Why winning imposes a halfspace constraint

For any valid upper bounds U_j, the true optimal validation score satisfies $v_* \le \min_j U_j = u$. Let h be the actual winner and write $\theta_h^* = c_h + e$. Convexity of V_h implies

$$V_h(c_h) + \nabla V_h(c_h)^\top e \le V_h(c_h + e) = v_* \le u.$$

Combining with $\|e\| \le \rho_h$ proves (9). If the resulting cap is empty, that candidate cannot be the winner. Nonemptiness is only a necessary condition. The cap can include points whose actual validation loss exceeds u because the tangent inequality is an outer approximation.

This distinction also explains why a cap need not contain the retained optimum of a *losing* candidate. Applying the cap to all candidates is safe for the selected-output enclosure, not for an unconditional claim about every candidate's prediction. In particular, the interval-information sharpness theorem is not applied to claim that these conditioned intervals are independent score/output rectangles.

D.2 Proof of the support formula

Derivation of Lemma 3; see Xiang et al. (2016). If $\rho = 0$, the only possible point is $e = 0$ and nonemptiness implies $b \ge 0$. Its support is zero. If $d = 0$, the halfspace either includes every point when $b \ge 0$ or is empty when $b < 0$. The support of the full ball is $\rho \|t\|$. These cases establish the degenerate statements. The direction $t = 0$ also has zero support whenever the region is nonempty.

Assume now $\rho > 0$, $d \ne 0$, and $t \ne 0$. Put $w = d/\|d\|$, $\alpha = b/\|d\|$, $t_\parallel = t^\top w$, and $t_\perp = \left\| t - t_\parallel w \right\|$. The range of $w^\top e$ on the ball is $[-\rho, \rho]$, so the cap is nonempty exactly when $\alpha \ge -\rho$.

The unconstrained ball maximizer is $e_0 = \rho t/\|t\|$, with value $\rho \|t\|$. It satisfies $w^\top e_0 = \rho t_\parallel / \|t\|$. Consequently, if $\rho t_\parallel \le \alpha \|t\|$, it is feasible and attains the first branch of (10).

Otherwise $w^\top e_0 > \alpha$, and necessarily $\alpha < \rho$. Decompose any feasible point as $e = zw + v$ with $w^\top v = 0$. Then

$$-\rho \le z \le \alpha, \qquad \|v\| \le \sqrt{\rho^2 - z^2},$$

and

$$t^\top e \le z t_\parallel + \sqrt{\rho^2 - z^2}\, t_\perp =: \psi(z).$$

For every admissible z, equality is attained by aligning v with $t - t_\parallel w$ when $t_\perp > 0$. When $t_\perp = 0$, any feasible v gives the same linear objective.

If $t_\perp > 0$, the function ψ is concave on $[-\rho, \rho]$, with its unique unconstrained maximizer at $z_0 = \rho t_\parallel / \|t\|$. It is nondecreasing to the left of z_0, as follows either from concavity or from

$$\psi'(z) = t_\parallel - \frac{z\, t_\perp}{\sqrt{\rho^2 - z^2}} \quad (-\rho < z < \rho).$$

Because $z_0 > \alpha$, the constrained maximum is $\psi(\alpha)$. If $t_\perp = 0$, the infeasible-unconstrained-maximizer condition forces $t_\parallel > 0$, so $\psi(z) = z t_\parallel$ is increasing and the maximum is again at α. The endpoint $\alpha = -\rho$ leaves the single point $e = -\rho w$ and satisfies the same formula by continuity. We obtain

$$s_C(t) = \alpha t_\parallel + \sqrt{\rho^2 - \alpha^2}\, t_\perp.$$

Finally, $\min_{e \in C} t^\top e = -s_C(-t)$, which proves the output interval. □

For $\alpha \ge \rho$, the first branch always applies because $t_\parallel \le \|t\|$. Thus the square root in the second branch is evaluated only where $-\rho \le \alpha < \rho$. The closed form requires a dot product, vector norms, and scalar arithmetic for each target once d and $\|d\|$ are known. It does not require fitting another model or solving a numerical constrained program.

E Adaptive safety, refinement complexity, and nontermination

E.1 Proof of the safety invariant

Proof of Theorem 4. Initially every candidate is active, so the actual selected candidate h_* is active. At any round in which h_* is active, valid score intervals imply

$$L_{h_*} \le v_* \le v_j \le U_j \quad \text{for every active } j.$$

The threshold u is therefore at least v_* and $L_{h_*} \le u$. For every earlier active index $j < h_*$, the prescribed tie rule gives $v_* < v_j$, so $L_{h_*} < U_j$. Thus h_* satisfies both parts of (6), and menu-aware pruning cannot remove it. Conditioning its output on $V_{h_*}(\theta^*_{h_*}) \le u$ is valid by Appendix D. Intersecting its new output region with any previous unconditional region or previous winning-conditional region preserves the actual output, since h_* is the winner of the same fixed counterfactual at every round.

This proves inductively that the selected candidate remains active and its true score and output remain enclosed. The original-output interval remains valid by assumption. The first part of Theorem 2 therefore encloses the actual attribution at every round. If a midpoint is returned, its error is at most half the enclosure width, coordinatewise for multiple outputs. If no width condition is met before the budget, returning only the enclosure preserves containment without asserting point accuracy. No part of the argument assumes a strict score gap or any particular refinement schedule. □

An empty winner-conditional region excludes the candidate under valid bounds, as formalized in Lemma 15. Rounding-induced emptiness is not a proof of exclusion. The implementation separately diagnoses contradictory unconditional bounds and eliminates candidates with empty conditional intersections. Its numerical guards do not prove real-arithmetic validity. An unresolved enclosure does not justify a point estimate.

E.2 Proof of the output-dependent work bound

Proof of Theorem 5. At the start of round zero no retained-data refinement has occurred. In Algorithm 1, every candidate that survives to round k has received exactly k refinements, because each surviving candidate is refined once between consecutive rounds. Previously computed interval intersections can only reduce widths.

Let h survive pruning at round k, and fix the actual tie-broken winner h_*. The safety theorem ensures that h_* is still active. Width at most Bq^k and containment imply

$$v_h \le L_h + Bq^k, \qquad U_{h_*} \le v_* + Bq^k.$$

The pruning rule gives $L_h \le \min_j U_j \le U_{h_*}$. Combining,

$$v_h - v_* \le 2Bq^k.$$

Every survivor thus belongs to the near-optimal set in (12).

Consider the unconditional output interval $[a_h^{\mathrm{unc}}, b_h^{\mathrm{unc}}]$ computed for that candidate at round k. It contains f_h and has width at most Aq^k, hence

$$a_h^{\mathrm{unc}} \ge f_h - Aq^k, \qquad b_h^{\mathrm{unc}} \le f_h + Aq^k.$$

The final possibly conditioned interval is a subset of this interval. Its endpoints satisfy the same one-sided inequalities even if it does not contain the output of a losing candidate. It follows that the hull of all surviving output intervals has width at most

$$\max_{h:v_h - v_* \le 2Bq^k} f_h - \min_{h:v_h - v_* \le 2Bq^k} f_h + 2Aq^k.$$

Subtracting the baseline interval adds exactly its width d_0. This proves (13), and the stated stopping conclusion follows directly from the midpoint rule.

To bound work, let $\delta_h = v_h - v_*$. If $2Bq^k < \delta_h$, then

$$L_h \ge v_h - Bq^k > v_* + Bq^k \ge U_{h_*}.$$

Any candidate still active at that round is removed by the pruning rule. Thus candidate h receives at most τ_h refinements before elimination, or at most T before the algorithm stops, whichever is smaller. Summing over candidates gives (14). If $\tau_h = 0$, its initial certificate eliminates it without a Newton refinement. If $\tau_h = \infty$, the minimum is T. This also resolves all zero-gap cases. □

For vector outputs, define

$$\omega_\infty(t) = \max_{1 \le r \le q} \left(\max_{h:v_h - v_* \le t} f_{hr} - \min_{h:v_h - v_* \le t} f_{hr} \right).$$

Assume every unconditional coordinate interval has width at most Aq_0^k, with contraction $q_0 \in (0,1)$, and every score interval has width at most Bq_0^k. Repeating the proof coordinatewise gives maximum attribution width at most $\omega_\infty(2Bq_0^k) + 2Aq_0^k + d_0$, where d_0 is the largest baseline coordinate width. We use q_0 here to avoid confusing the contraction with the number of targets.

E.3 Separation from winner identification

Proposition 10 (An interval-oracle separation)**.** *Consider two candidates with true scores $v_1 = 0$, $v_2 = \delta > 0$ and true outputs $f_1 = f_2 = 0$. At refinement level k, an oracle supplies score intervals $[v_h - Bq^k, v_h + Bq^k]$ and output intervals $[-Aq^k, Aq^k]$, with known baseline zero. Take $A, B > 0$, $0 < q < 1$, and $\varepsilon > 0$. An attribution midpoint is valid to tolerance ε whenever $Aq^k \le \varepsilon$. Winner identity remains ambiguous under the score intervals whenever $\delta < 2Bq^k$.*

Proof. Both output intervals are $[-Aq^k, Aq^k]$. Their hull is the same interval whether one or two candidates are feasible, so its midpoint zero has certified error Aq^k. For candidate two to be a feasible winner when candidate one wins ties, Theorem 2 requires $\delta - Bq^k < Bq^k$, equivalent to $\delta < 2Bq^k$. Candidate one is always feasible. Thus both candidate identities remain possible under the score intervals in this regime. □

For any prescribed finite output-certification round, δ can be chosen small enough that winner identity remains unresolved for arbitrarily many additional refinements. This is an information separation for the specified interval oracle, not a lower bound on every algorithm that can read a training dataset. A method exploiting additional exact algebraic relationships can receive more information than the oracle supplies.

Conversely, if two feasible winners can have outputs separated by more than 2ε under independent score/output information, no scalar estimate has worst-case error at most ε without narrowing the feasible output set. This follows from the endpoint lower bound in Appendix B. Refinement must then improve relevant output bounds, eliminate a candidate, or exploit structure that couples scores and outputs.

E.4 Exact ties and sufficient termination conditions

Suppose there is a unique winner with gap $\gamma = \min_{h \neq h_*}(v_h - v_*) > 0$. Once $2Bq^k < \gamma$, every nonwinner is eliminated. If also $2Aq^k + d_0 \leq 2\varepsilon$, the output is certified. Thus geometric refinement and $d_0 < 2\varepsilon$ imply finite termination in the unique-winner case.

For a finite menu with tied minimizers, there is still a positive gap to the nontied candidates unless all candidates tie. After excluding every positive-gap candidate, the spread bound depends on the outputs of the tied candidates. If that spread plus d_0 is strictly below 2ε, sufficiently tight output intervals give finite termination. If tied outputs are farther apart, a conservative interval procedure may fail to terminate despite a deterministic tie rule specifying the actual winner. Intervals approaching equal scores do not necessarily prove their exact equality at any finite stage. Symbolic equality information or an exact score evaluation could resolve this issue, but is not assumed by the algorithm. A valid unresolved interval is the appropriate output when such information is unavailable.

E.5 What the step bound counts

Equation (14) counts refinement steps only. It excludes full-data bank construction, warm starts, and initial candidate certification. For a dense logistic problem with retained sample size m and dimension p_h, forming a Hessian costs $O(mp_h^2)$ and its Cholesky factorization costs $O(p_h^3)$. Gradient evaluation costs $O(mp_h)$, and an objective line search adds one such pass per trial. Validation value and gradient evaluation cost $O(m_v p_h)$, and q linear outputs cost $O(qp_h)$. The validation smoothness bound is precomputed from the fixed validation design. An influence warm start uses one retained gradient and one solve with a cached full-data factorization for each candidate.

Eliminating a candidate avoids its later training and certificate passes, but it does not refund its initialization cost. A ratio of Newton counts therefore need not equal the runtime ratio. Our matched-tolerance ablation times initialization and certification for both allocation policies, and the main text reports measured time separately from counted refinements.

F Exact normalized ridge deletion and a certified refinement ladder

Let the full design be $X \in \mathbb{R}^{n \times p}$ with responses y, and delete $s \in \{1, \ldots, n-1\}$ rows $U = X_G$ with responses y_G. The retained ridge solution minimizes

$$\frac{1}{2(n-s)} \left\| X_{-G}\theta - y_{-G} \right\|^2 + \frac{\lambda}{2} \left\| \theta \right\|^2, \qquad \lambda > 0.$$

Its normal matrix and right-hand side are

$$P = X^\top X - U^\top U + (n-s)\lambda I, \qquad b^- = X^\top y - U^\top y_G.$$

Set $A = X^\top X + (n-s)\lambda I$, $B_G = A^{-1}U^\top$, $E = UB_G$, $\bar{\theta} = A^{-1}X^\top y$, and $r = U\bar{\theta} - y_G$.

Lemma 11 (Group-space deletion identity)**.** *The matrix E is symmetric positive semidefinite with all eigenvalues below one. The retained ridge solution is*

$$\theta^* = \bar{\theta} + B_G(I - E)^{-1}r.$$

Proof. Both A and P are positive definite since $(n-s)\lambda > 0$. Let $C = UA^{-1/2}$. Then $E = CC^\top \succeq 0$, while

$$A^{-1/2}PA^{-1/2} = I - C^\top C \succ 0.$$

Every eigenvalue of $C^\top C$ is strictly below one. The nonzero eigenvalues of $CC^\top$ and $C^\top C$ agree, and any additional eigenvalues are zero. Hence $0 \preceq E \prec I$, so $I - E$ is invertible.

The identities

$$P\bar{\theta} = X^\top y - U^\top U\bar{\theta}, \qquad PB_G = (A - U^\top U)A^{-1}U^\top = U^\top(I - E)$$

imply

$$\begin{aligned} P\big(\bar{\theta} + B_G(I-E)^{-1}r\big) &= X^\top y - U^\top U\bar{\theta} + U^\top r \\ &= X^\top y - U^\top y_G = b^-. \end{aligned}$$

Since P is positive definite, its normal equation has a unique solution, proving the claim. □

The anchor $\bar{\theta}$ is generally not the full-data solution at the original mean-loss coefficient λ. Its regularization term in the normal equation is $(n-s)\lambda I$, as required by the retained normalization. Substituting $n\lambda I$ without an additional correction would produce the wrong counterfactual.

Theorem 12 (Ridge output and validation intervals)**.** *For $k \geq 0$, define*

$$\theta^{(k)} = \bar{\theta} + B_G \sum_{j=0}^{k-1} E^j r, \qquad r_k = E^k r, \qquad \eta_k = \frac{\|r_k\|}{1-\bar{\rho}},$$

where the empty sum is zero and $\|E\|_2 \leq \bar{\rho} < 1$. For any linear target t,

$$|t^\top(\theta^* - \theta^{(k)})| \leq \left\|t^\top B_G\right\| \eta_k.$$

For a validation design $Z \in \mathbb{R}^{m_v \times p}$, response y_v, and validation loss $V(\theta) = \|Z\theta - y_v\|^2/(2m_v)$, put $\zeta_k = \|ZB_G\|_2 \eta_k$ and $s_k = \left\|Z\theta^{(k)} - y_v\right\|$. Then

$$\frac{\max\{0, s_k - \zeta_k\}^2}{2m_v} \leq V(\theta^*) \leq \frac{(s_k + \zeta_k)^2}{2m_v}. \tag{20}$$

Proof. Because $\|E\|_2 < 1$, the Neumann series converges in operator norm. Lemma 11 yields

$$\theta^* - \theta^{(k)} = B_G \sum_{j=k}^{\infty} E^j r = B_G(I-E)^{-1} r_k.$$

The inverse has operator norm at most $1/(1-\bar{\rho})$, since E is symmetric positive semidefinite. Thus $\left\|(I-E)^{-1}r_k\right\| \leq \eta_k$. Multiplying by $t^\top B_G$ and applying Cauchy–Schwarz proves the target bound. Multiplying by ZB_G gives $\left\|Z(\theta^* - \theta^{(k)})\right\| \leq \zeta_k$. The direct and reverse triangle inequalities imply

$$\max\{0, s_k - \zeta_k\} \leq \|Z\theta^* - y_v\| \leq s_k + \zeta_k.$$

Squaring nonnegative terms and dividing by $2m_v$ gives (20). □

The quantities r_k satisfy $r_{k+1} = Er_k$ and $\|r_k\| \leq \bar{\rho}^k \|r\|$. Thus the output width decays geometrically. For validation, the norm of $Z\theta^{(k)} - y_v$ is bounded uniformly in k by $\|Z\theta^* - y_v\| + \|ZB_G\|_2 \|r\|/(1-\bar{\rho})$. If w_k is the interval width in (20), a case split on $s_k \geq \zeta_k$ gives

$$w_k \leq \frac{2s_k\zeta_k}{m_v} + \frac{\zeta_k^2}{2m_v}.$$

Indeed, the width is $2s_k\zeta_k/m_v$ in the first case, while in the second case $s_k^2 \leq 2s_k\zeta_k$ bounds $(s_k+\zeta_k)^2/(2m_v)$. Thus the width is bounded by a constant times $\bar{\rho}^k$ plus a constant times $\bar{\rho}^{2k}$, and hence by a constant times $\bar{\rho}^k$. This supplies the geometric-width premise of Theorem 5 whenever a common $\bar{\rho} < 1$ bounds the finite candidate menu. The special case $E = 0$ is exact after one term and does not require a nonzero contraction.

F.1 Initialization and refinement costs

Constructing $X^\top X$ and $X^\top y$ costs $O(np^2)$ and $O(np)$ once. For a fixed pair (λ, s), the factorization of A and the anchor are shared across all groups of size s. Without such a cache, factorization costs $O(p^3)$ for each pair. For a particular group, obtaining B_G costs $O(p^2 s)$ and forming E costs $O(ps^2)$. An exact eigendecomposition for $\|E\|_2$ costs $O(s^3)$. The implementation pays this cost. A cheaper valid bound, such as $\|E\|_\infty$ for symmetric E, can be substituted when it is below one, but need not be tight enough to be useful.

After forming the target map TB_G and validation map ZB_G, a new tail vector requires $O(s^2)$ work, and updating all target and validation predictions costs $O((q + m_v)s)$. The map construction, their norm bounds, and the initial certificate are not free. An exact solve with $I - E$ also costs $O(s^3)$ plus lower-order work, which may be competitive with computing a spectral certificate. The ridge identity therefore provides an exact deletion-refinement procedure, but need not be faster than a direct group-space solve. Our ridge experiments are correctness checks, not a speed benchmark.

G Pre-refit certificates and the Newton solver contract

Residual validity, Newton refinement, and convergence rates require different assumptions. A residual region is valid for any differentiable strongly convex objective. Newton refinement requires a Hessian, and its quantitative rate additionally requires a bound on Hessian variation. A residual certificate or a finite numerical audit does not supply this rate assumption.

G.1 A certificate before any retained-data optimization

For each candidate choose an available full-data approximation $\bar{\theta}_h$. Evaluate the retained residual using (16), including the original optimization residual, or evaluate it directly on the retained sample. The resulting shifted ball (7) and score/output intervals (8) do not require a retained-data Newton step.

Corollary 13 (Pre-refit no-switch and output-agreement tests)**.** *Suppose all these intervals are valid. If a candidate h satisfies*

$$U_h < L_j \quad (j < h), \qquad U_h \le L_j \quad (j > h),$$

then h is the selected retained-data candidate. In particular, if h is a certified original winner, these inequalities certify that selection does not change. Whether or not the winner is identified, the hull over the tie-feasible candidates supplies an attribution estimate before retained-data optimization whenever its width, including baseline uncertainty, is at most 2ε.

Proof. For an earlier candidate, $v_h \le U_h < L_j \le v_j$. For a later candidate, $v_h \le U_h \le L_j \le v_j$. Therefore every rival either has larger score or loses an exact tie to h. The second claim is Theorem 2 applied to the valid pre-refit intervals. The tests are sufficient, not necessary; a failed test does not establish that a switch occurs or that a refit is unavoidable for every algorithm. □

The test includes the cost of residual and validation evaluation. Reusing a stored full-data residual can avoid a retained-sample gradient pass through (16), but storing or certifying that residual has a cost. No saving is asserted when those operations cost more than a direct refit.

G.2 Full and damped Newton residuals

Let R be twice continuously differentiable and μ-strongly convex, with $\mu > 0$. At θ write $r = \nabla R(\theta)$ and $H = \nabla^2 R(\theta)$. Let p be a computed displacement, and let $\xi = Hp + r$ be its linear-solve residual. Assume the Hessian is M-Lipschitz in operator norm on the segment $\{\theta + tp : 0 \le t \le 1\}$. Then

$$\|\nabla R(\theta + ap)\| \le (1 - a)\,\|r\| + a\,\|\xi\| + \frac{Ma^2}{2}\,\|p\|^2, \qquad 0 \le a \le 1. \tag{21}$$

Indeed, integration along the segment gives

$$\nabla R(\theta + ap) = r + aHp + \int_0^a \big(\nabla^2 R(\theta + tp) - H\big)p\,dt.$$

Substitute $Hp = -r+\xi$. The integral has norm at most $M\|p\|^2 \int_0^a t\,dt = Ma^2\|p\|^2/2$, proving (21). Strong convexity implies $H \succeq \mu I$ and therefore

$$\|p\| \le \frac{\|r\| + \|\xi\|}{\mu}.$$

Proposition 14 (Conditional local residual contraction)**.** *Suppose every subsequent step is a full step, every relevant segment satisfies the same Hessian-Lipschitz bound M, and each solve has $\|\xi\|_k \le \eta\|r\|_k$ for a fixed $0 \le \eta < 1$. If*

$$q = \eta + \frac{M(1+\eta)^2}{2\mu^2}\|r\|_0 < 1,$$

then $\|r\|_k \le q^k\|r\|_0$ for all such steps. In the exact-solve case, one full step satisfies

$$\|r\|_{k+1} \le \frac{M}{2\mu^2}\|r\|_k^2.$$

Proof. Taking $a = 1$ in (21) yields

$$\|r\|_{k+1} \le \left(\eta + \frac{M(1+\eta)^2}{2\mu^2}\|r\|_k\right)\|r\|_k.$$

At $k = 0$ the bracket is $q < 1$. If $\|r\|_k \le q^k\|r\|_0$, the bracket remains at most q. Induction proves the geometric bound. Setting $\eta = 0$ gives the last display. □

The segment assumption is part of the proposition. It must hold on a region containing the iterates and the step segments, or globally. The proposition does not prove that a line search eventually accepts full steps. For damped steps (21) remains valid, but a uniform contraction factor requires additional conditions on the accepted step sizes and residuals.

G.3 From a residual rate to interval-width envelopes

Suppose the preceding residual contraction holds for a candidate and put $C = \|r\|_0/(2\mu)$. Its shifted-ball radius obeys $\rho_k \le Cq^k$. For a linear output $t^\top\theta$, the unconditional interval width is at most

$$Aq^k, \qquad A = 2\|t\|\,C.$$

Suppose also that V is convex and β-smooth on every parameter ball used, and that $\|\nabla V(c_k)\| \le D$ at their centers. Then (8) has width at most

$$2DCq^k + \frac{\beta C^2}{2}q^{2k} \le Bq^k, \qquad B = 2DC + \frac{\beta C^2}{2}.$$

The bound on the center gradients is a separate condition. For logistic validation it follows globally from $\|\nabla V(c)\| \le m_v^{-1}\sum_i \|z\|_i$, because the scalar logistic derivative has magnitude at most one. The logistic Hessian-Lipschitz constant in Appendix C is likewise global. On a finite menu, enlarge the candidate-specific constants to common A, B and take the largest $q < 1$, provided every candidate satisfies the local full-step conditions. These choices give a conditional application of Theorem 5, not an unconditional rate for the empirical damped solver.

G.4 Operational outputs and unresolved numerical ties

For true-label probability obtained from a signed logit, $\sigma'(z) \le 1/4$. A valid signed-logit interval $[a, b]$ therefore gives probability interval $[\sigma(a), \sigma(b)]$ of width at most $(b-a)/4$. If the target is a probability change, transform the post-deletion and baseline intervals separately before subtracting. An error bound for the difference of logits alone does not justify applying the sigmoid to that difference. A positive lower endpoint or a negative upper endpoint certifies the sign. An interval straddling zero does not. These statements concern the fixed computational target, not predictive-probability calibration.

Two exactly equal score intervals $[v, v]$ with different exact outputs are resolved by the menu order. Two narrow but overlapping floating-point intervals do not establish exact equality. The

implementation must keep all feasible winners unless the valid intervals or a separate exact argument exclude them. High numerical precision is not an exact comparison oracle. Returning an unresolved enclosure is appropriate when the available numerical information cannot establish the requested accuracy.

H CONDITIONAL INFEASIBILITY AND ACCUMULATED INFORMATION

A candidate's validation score and output are fixed numbers after the deletion is specified. The premise that this candidate wins is also fixed. Refinement changes the information available about these quantities, rather than changing the counterfactual.

Let $J_h^{(k)}$ be a valid unconditional output box after computation k, and let $P_h^{(k)}$ be an output box known to contain f_h whenever $h = h_*$. Define

$$Q_h^{(k)} = \bigcap_{\ell \le k} J_h^{(\ell)} \cap \bigcap_{\ell \le k} P_h^{(\ell)}.$$

Lemma 15 (Conditional contradictions exclude candidates)**.** *If all stated bounds are valid and $Q_h^{(k)} = \varnothing$, then $h \neq h_*$. The true winner therefore survives score pruning, deterministic tie pruning, and conditional-empty-set pruning at every round. Intersecting score intervals or unconditional output intervals cannot itself yield an empty set under these assumptions.*

Proof. If $h = h_*$, its fixed output f_h belongs to every unconditional box and, by the common winning premise, every conditional box. It therefore belongs to $Q_h^{(k)}$. An empty intersection contradicts $h = h_*$. This argument applies at every adaptively chosen computation because the bounds are assumed valid whenever evaluated. The score and tie arguments in Theorem 4 then preserve the same winner. An unconditional enclosure, unlike a winner-conditional enclosure, must contain the output even when the candidate loses. All unconditional intersections consequently contain that fixed output. □

The rule is sufficient, not necessary. Coordinatewise boxes may overlap even when the underlying parameter regions do not. A single empty dome supplies no exclusion beyond its own tangent-score lower bound. Intersecting information from different computations, however, can reveal a contradiction that neither scalar lower bound detects alone.

H.1 A SMOOTH REGRESSION CASE

The following smooth example isolates the distinction and is included as a regression test. Candidate one has exact validation score $1/2$ and exact output zero. Candidate two has

$$R(\theta) = \tfrac{1}{2}\|\theta\|^2, \qquad V(\theta) = \tfrac{1}{2}\|A\theta - b\|^2, \qquad g(\theta) = (-0.2, -0.14)^\top \theta,$$

where

$$A = \begin{pmatrix} -1.6 & -0.1 \\ 1.1 & 0.3 \end{pmatrix}, \qquad b = \begin{pmatrix} -0.2 \\ 1.5 \end{pmatrix}.$$

The optimum of the second training objective is zero, with validation score 1.145, so it loses. An exact Newton displacement damped by 0.8 changes $(1.4, -4.2)$ to $(0.28, -0.84)$ and decreases the objective by 96%. Its shifted-ball score intervals are approximately $[0.085279, 11.662447]$ and $[0.328007, 2.214863]$. Both permit the candidate to beat $1/2$ according to scalar score information. The corresponding conditional output intervals are approximately

$$[-0.386407, -0.117862], \qquad [-0.077281, -0.024272].$$

They are disjoint by more than 0.04. Their intersection excludes candidate two, leaving the exact output zero. Neither unconditional interval is contradicted. Widening a numerical tolerance would miss the distinction between invalid evidence and an impossible winning premise. The test follows the prescribed damped trajectory; it does not assume that the logistic Armijo implementation would choose it.

H.2 NUMERICAL CONTRACT

The implementation keeps unconditional score intervals, unconditional output intervals, and winner-conditional output intervals in separate arrays. It rejects nonfinite or reversed unconditional endpoints and inconsistent unconditional intersections. It removes a candidate whose accumulated conditional intersection is empty. If all winning premises become infeasible, it raises a diagnostic error rather than returning an estimate. The implementation uses ordinary float64 arithmetic and explicit numerical guards. The checks implement the algorithm's logical distinctions. They do not prove that every computed endpoint encloses its real-arithmetic target.

I PROOF OF THE FINITE-DATA SEPARATION

We give the full calculation for Proposition 6. The sample contains three observations in two dimensions. Squared loss is $\ell((x,y),\theta) = (x^\top\theta - y)^2/2$. The covariance and cross moment before deletion are

$$\frac{X^\top X}{3} = \mathrm{diag}(1, 2/3), \qquad \frac{X^\top y}{3} = (1/3, 2/3)^\top.$$

After deletion they are I_2 and e_2. Thus

$$\theta_\lambda(D) = \left(\frac{1}{3(1+\lambda)}, \frac{2}{2+3\lambda}\right)^\top, \qquad \theta_\lambda(D^-) = \left(0, \frac{1}{1+\lambda}\right)^\top. \tag{22}$$

The retained validation predictions are $1/2$ and $1/4$. Their score difference is

$$v_2 - v_1 = \tfrac{1}{2}(1/4 - 3/8 - \delta)^2 - \tfrac{1}{2}(1/2 - 3/8 - \delta)^2 = \delta/4 > 0.$$

The full-data validation predictions are $2/5$ and $2/11$. Their midpoint is $16/55 < 3/8$, so candidate one also wins before deletion for every admitted δ. Its original output is $1/6$ and its retained output is zero. Therefore $\Delta_G = -1/6$, independently of δ.

I.1 THE ITERATES AND THEIR RESIDUAL BALLS

The full Hessian is $H_\lambda(D) = \mathrm{diag}(1+\lambda, 2/3+\lambda)$. Substituting (22) into the normalization-aware warm start gives

$$\widetilde{\theta}_\lambda^{(0)} = \theta_\lambda(D^-) + d_\lambda e_2, \qquad d_\lambda = \frac{\lambda}{(1+\lambda)(2+3\lambda)^2}.$$

In particular $d_1 = 1/50$ and $d_3 = 3/484$. The retained Hessian is $(1+\lambda)I_2$. A half Newton step halves the error, so after k refinements

$$\widetilde{\theta}_\lambda^{(k)} = \theta_\lambda(D^-) + 2^{-k} d_\lambda e_2, \quad r_\lambda^{(k)} = (1+\lambda)2^{-k} d_\lambda e_2.$$

The chosen strong-convexity lower bound is $\mu_\lambda = \lambda$, not the sharper $1+\lambda$. It gives

$$c_\lambda^{(k)} = \theta_\lambda(D^-) + \frac{\lambda - 1}{2\lambda}\, 2^{-k} d_\lambda e_2,$$
$$\rho_\lambda^{(k)} = \frac{2^{-k}}{2(2+3\lambda)^2}.$$

Both centers have first coordinate zero. Candidate one's radius is $2^{-k}/50$ and candidate two's is $2^{-k}/242$. Candidate one's center equals its optimum. The unconditional output interval of candidate one consequently contains that of candidate two. Since the true winner remains active, the selected-output hull, after subtracting the exact baseline, is

$$I_G^{(k)} = \left[-\frac{1}{6} - \frac{2^{-k}}{50}, -\frac{1}{6} + \frac{2^{-k}}{50}\right].$$

This remains the hull if the loser has already been removed. Intersecting historical output intervals does not change it because these intervals are nested. The midpoint condition is exactly $2^{-k}/50 \le \varepsilon$, proving the output-refinement upper bound in Proposition 6. At most two candidates are refined in each preceding round.

I.2 WHY THE WINNER RACE REQUIRES MORE ROUNDS

Write $\rho_k = 2^{-k}/50$. Candidate one's validation gradient at its center has norm $1/8 - \delta \geq 1/16$ and validation smoothness is one. Its upper score endpoint is

$$U_1^{(k)} = v_1 + (1/8 - \delta)\rho_k + \rho_k^2/2 \geq v_1 + \rho_k/16.$$

Its historical upper endpoints decrease, so accumulation preserves this current upper bound. Candidate two's accumulated lower endpoint is at most its true score $v_2 = v_1 + \delta/4$. If $2^{-k} > 200\delta$, then $\rho_k > 4\delta$ and therefore

$$L_2^{(k)} \leq v_2 < U_1^{(k)}.$$

Candidate two remains feasible under the strict earlier-menu comparison. Candidate one remains feasible because it is the true winner. Hence both candidates are refined at every round preceding $k_\delta = \lceil \log_2(1/(200\delta)) \rceil$. A score-pruning race required to identify the winner has performed at least $2k_\delta$ candidate refinements before it can terminate. This comparator uses rectangular output bounds, the same accumulated score bounds, and the same synchronous half-step schedule. The statement does not cover a comparator that solves the normal equations directly or reasons from other properties of the shared data.

A full Newton step sends either candidate to its retained optimum. Every residual region then has zero radius in real arithmetic. The two distinct validation scores identify the winner and the output is known exactly. Both races terminate after at most one full step per candidate. The control shows why a separation under gradual refinement can disappear when the solver reaches the required precision in a single step.

I.3 A CONTROLLED OUTPUT-DISAGREEMENT AXIS

Replace the output by

$$g_\eta(\theta) = \frac{\theta_1 + \eta\theta_2}{\sqrt{1+\eta^2}}, \qquad \eta \geq 0.$$

The query direction remains a unit vector. The retained output difference is exactly

$$|f_1 - f_2| = \frac{\eta}{4\sqrt{1+\eta^2}},$$

whereas the validation gap remains $\delta/4$. Whenever both candidates are retained, any valid *unconditional* output hull has at least this width. Rectangular output stopping is therefore impossible at $2\varepsilon < \eta/(4\sqrt{1+\eta^2})$ while both remain feasible. This necessity claim is for unconditional output information. Conditional bounds can instead rule out the losing premise and need not contain its true output. The experimental sweep varies η, δ, tolerance, and damping independently, including the regions where no additional output-stopping benefit is possible.

J EXPERIMENTAL PROTOCOL AND IMPLEMENTATION

Execution and source version. All empirical tables use the current execution records. The 4,800 main queries, 360 dense-menu evaluations, and 600 allocation queries were rerun with `research/code/core.py`. This implementation handles deterministic ties and distinguishes conditional infeasibility from contradictory unconditional evidence. The original archive is retained separately for provenance. Appendix M describes the additional studies and checks.

J.1 DATASETS, SPLITS, AND TARGETS

The public datasets are accessed through the scikit-learn loaders `load_breast_cancer`, `load_wine`, and `load_digits` (Pedregosa et al., 2011). Their total sample sizes are 569, 178, and 1,797, with 30, 13, and 64 raw features, respectively. The breast-cancer task retains the loader's binary labels. The positive class in wine is the loader's first class; the remaining classes are negative. For digits, the upper half of the ten digit classes is positive and the lower half is negative. These are the rules in `prepare` in `research/code/run_experiments.py`. No rows are removed on the basis of loss, leverage, selection margin, or attribution magnitude.

For seeds $r = 0, \ldots, 9$, we call `train_test_split` with test fraction 0.4, random state r, and stratification by the binary target. The remainder is split equally into validation and test sets with random state $r + 101$, again stratified by the binary target. The first 16 examples in the returned test ordering define the attributed signed logits. Their identifiers are fixed within each split and are saved in the metadata. All 16 outputs are considered simultaneously for the stopping rule.

For each raw-feature or frozen-feature representation, `StandardScaler` is fitted to the original head-training sample. Transformed nonconstant coordinates are divided by the square root of the representation dimension, and a constant intercept coordinate is appended. The scaler is not refitted after deletion. Both the training loss and validation loss use the resulting fixed feature representation. All head parameters, including the intercept, receive the same ridge penalty. For neighborhood deletions, distances are instead computed in the standardized *raw* training features divided by the square root of the raw dimension. This choice is also fixed in the neural-feature experiment.

Table 3: Sample sizes and dimensions after preprocessing. The last three columns are mean measured milliseconds. Bank time covers the logistic model bank and Hessian factorizations, excluding feature preparation and upstream neural training. Reference time covers influence initialization and retained-model optimization, excluding its subsequent selection verification. Query timing scopes are detailed below.

Setting	Train	Valid.	Test	p	K	Bank ms	SWA ms	Ref. ms
Breast cancer	341	114	114	31	19	17.5	4.84	6.33
Wine	106	36	36	14	19	10.5	3.70	3.82
Digits	1078	359	360	65	19	68.4	18.57	27.77
Frozen checkpoints	808	269	270	33	39	47.7	14.72	21.02

J.2 Frozen checkpoint experiment

The neural experiment first holds out 25% of the complete digits dataset as an upstream training set, using random state $3000 + r$ and stratification by the ten digit labels. The remaining 75% is the head-data pool, which is split into training, validation, and test sets as above. The upstream pool is disjoint from all head-training, validation, and test examples.

A multilayer perceptron has widths $64 \to 64 \to 32 \to 10$, with ReLU after each hidden layer. Pixel inputs are divided by 16. We train for 40 epochs on the upstream data with ten-class cross-entropy and Adam (Kingma & Ba, 2015), learning rate 0.01, batch size 64, and the library defaults for the other optimizer parameters. PyTorch is seeded with $7000 + r$, and the batch-permutation generator with $8000 + r$. Deterministic algorithms and one computational thread are requested. The 32-dimensional hidden features after epochs 5, 15, and 40 are extracted without gradients, converted from float32 to float64, and standardized using the original head-training sample. The candidate menu is the Cartesian product of these three representations and the 13-value regularization grid.

Deletion affects only the binary logistic head-training sample. The feature extractor, its upstream sample, checkpoint times, and learned weights are not changed. Experiment seeds change both the upstream split and the training seed. The validation sample also changes, so the reported seed variation does not isolate variability from upstream initialization. Only the final neural weights are stored in the raw outputs, while the fixed program and seeds regenerate the earlier feature snapshots.

J.3 Candidate menus and group generation

For raw features, the grid is `logspace(-6,0,19)`. For frozen checkpoints, each representation uses `logspace(-6,0,13)`. Candidates are ordered first by checkpoint, when applicable, and then by increasing regularization value. NumPy's first-minimum behavior implements the menu-order tie rule. Main numerical references are additionally checked for strict score separation, so no exact reference tie occurs in the main experiment.

Each setting and seed has six intervention conditions, formed by fractions $f \in \{0.01, 0.05, 0.10\}$ and mechanisms uniform and neighborhood. Each condition has group identifiers $j = 0, \ldots, 19$. The requested size is $s = \max\{1, \mathrm{round}(fn)\}$, using Python's nearest-integer rounding. The actual retained objective always uses the integer size. Query identifiers run from zero to 119, with

fraction as the outer loop, then mechanism, then group identifier. The NumPy generator seed is $100000 + 10000r + \text{query_id}$.

Uniform deletion samples s indices without replacement. Neighborhood deletion samples a center uniformly from the head-training sample, computes squared Euclidean distances in the fixed standardized raw feature space, and deletes the s nearest indices, using a stable sort. Groups are sorted before saving. Neighborhoods define feature-space interventions. They do not establish shared real-world provenance. All groups are specified before inspecting post-deletion validation losses. The saved CSV files list every deleted index and the surviving candidate indices.

J.4 Optimization and reference construction

The logistic objective, gradient, and Hessian are evaluated directly in double precision, using `logaddexp` for the loss and the stable sigmoid implementation in SciPy. A dense Cholesky factorization gives the Newton direction. The line search begins at step length one and halves the step until the Armijo condition with coefficient 10^{-4} is satisfied. It permits a rounding allowance of $2 \times 10^{-15} \max\{1, |R(\theta)|\}$ and raises an error after 40 unsuccessful trials.

Full-data candidates are initialized at zero. Reference fits terminate when

$$\frac{\|\nabla R_h(\theta)\|}{\lambda_h} \le 10^{-9},$$

with a limit of 100 Newton refinements. Failure to attain this tolerance raises an error. Full-data Hessians at the fitted models are factorized and cached for all deletion queries on that split.

For each group, the retained gradient at each original model gives its influence warm start. Every tight-reference candidate starts from that same approximation and is fitted to the 10^{-9} parameter-radius tolerance. Validation loss selects the reference winner. Shifted-ball validation intervals at the fitted candidates must strictly separate both the original and the main post-deletion winner. A failure of numerical separation terminates the run rather than silently choosing a supposedly exact winner. All main queries completed.

If the original selected model has residual parameter radius r_0, its output coordinate interval is centered on $t_0^\top \widehat{\theta}_0$ with radius $\|t_0\| r_0 + 10^{-12}$. The retained reference output adds its own residual-derived coordinate radius. The largest combined reference radius over all main queries is 4.92×10^{-9}, rounded upward. This reference uncertainty is much smaller than 10^{-3} but is not a directed-rounding guarantee.

J.5 Baseline formulas and common initialization

Write H_h for the cached full-data Hessian, $\widehat{\theta}_h$ for the original candidate, and $r_h^- = \nabla R_h(\widehat{\theta}_h; D^-)$. The influence approximation is

$$\theta_h^{\mathrm{IF}} = \widehat{\theta}_h - H_h^{-1} r_h^-.$$

Fixed IF evaluates this approximation at h_0. *Reselect IF* chooses the candidate minimizing the actual validation loss at θ_h^{IF}. *Fixed exact* instead uses the tightly refitted retained parameter of h_0.

Reselect Newton takes one damped Newton step for the retained objective at $\widehat{\theta}_h$ for every candidate. *IF + Newton* takes one such step at θ_h^{IF}. Both then choose the candidate by the validation loss at the resulting approximation. Their output attribution subtracts the same original selected output. We do not add a Taylor approximation of validation loss on top of their parameter approximation.

The SWA candidate states also start at θ_h^{IF}. They use $\mu_h = \lambda_h$, the precomputed validation smoothness constant (19), intersected score/output intervals, and the selection-conditioned cap. All surviving candidates receive one Newton step per round. The main tolerance is $\varepsilon = 10^{-3}$, with at most 50 rounds. An unresolved computation returns no point estimate, and all main computations resolved.

J.6 Restricted linear TRAK-style ranking baseline

The ranking comparator uses only the original selected model and its fixed features. Let $s_i = 2y_i - 1$, $\phi_i = s_i x_i$, and $q_i = \sigma(-s_i x_i^\top \widehat{\theta}_{h_0})$. Put the row vectors $\phi_i^\top$ in Φ, and let $t = s_{\mathrm{test}} x_{\mathrm{test}}$ be a signed

test feature. The implemented score for a deletion group is

$$a_G^{\mathrm{linTRAK}}(t) = -\sum_{i \in G} q_i \, t^\top (\Phi^\top \Phi + \gamma I)^{-1} \phi_i, \qquad \gamma = 10^{-6} \frac{\mathrm{tr}(\Phi^\top \Phi)}{p}. \tag{23}$$

The sign converts a summed training-contribution score to the deletion convention. This full-feature, single-model linear specialization is motivated by TRAK (Park et al., 2023). It omits the original method's random projections and multi-model aggregation. The checkpoint menu is not a TRAK ensemble. We evaluate this score only by ranking correlation, not as a calibrated finite-deletion effect or a benchmark of the complete TRAK system.

For each dataset, seed, fraction, mechanism, and one of 16 targets, we compute Spearman correlation across the 20 deletion groups. Correlations are first averaged within a seed, then summarized across seeds. None of the implemented correlations is undefined. We use the same grouping for all other ranking methods.

J.7 Ablations and timing

The matched-allocation ablation uses seeds zero through four and group identifiers zero through four in each of the six main conditions. This selection depends only on identifiers, giving 150 queries per setting and 600 overall. The uniform comparator evaluates and refines every candidate each round, including candidates already excluded as winners. It uses the same surviving-output hull, selection conditioning, baseline interval, and 10^{-3} tolerance. The comparison therefore changes allocation of later work, not point accuracy or the stopping requirement. Original model preparation is repeated with the same seeds.

A separate dense-grid ablation uses breast cancer and digits, seeds zero through two, 10% deletion, both mechanisms, and five groups per condition. Its generator seed is $99000 + 100r + j$. The same deletion case is evaluated for $K \in \{65, 257\}$ logarithmically spaced values from 10^{-6} to 1 and $\varepsilon \in \{10^{-3}, 10^{-2}, 10^{-1}\}$. There are 60 dataset/seed/mechanism/group cases, 120 case/menu pairs, and 360 case/menu/tolerance runs. These are not 360 independent deletions. Tight reference fits use the same parameter tolerance as the main study. This ablation applies the same strict interval-separation assertion to every post-deletion reference winner. Its smallest recorded gap is approximately 1.2410×10^{-7}.

The dense variants are SWA, a rectangular-output variant with cap conditioning removed, and a rectangular-output variant additionally required to have one surviving candidate. All other refinement and stopping settings agree. Every run resolves and meets its requested error tolerance against the tight numerical reference.

Main reference and SWA execution order alternates with query parity. The matched-allocation study alternates uniform and adaptive execution order using seed and query parity. Each method recomputes its influence initialization within its own timer. A separate repeated four-order timing study is described in Appendix M; no cross-machine performance study is claimed. Newton counts remain the primary allocation comparison.

A main SWA timer includes constructing the retained candidate states, computing validation and output certificates, pruning, and all refinements, plus the common influence initialization. A main reference timer includes common initialization and retained-model optimization, but excludes the subsequent validation selection and its interval-verification audit. The stored *Reselect IF* timer records initialization without its separately evaluated validation-selection pass. These timers should not be interpreted as identically instrumented end-to-end comparisons across all baselines. In contrast, both matched-allocation timers enclose candidate construction and the complete same-certificate race, including each method's separately measured initialization. Full-data bank time and upstream feature preparation are excluded from every per-deletion timer. The bank is an amortized cost shared across all 120 queries of a split.

J.8 Environment and commands

The executed environment used Python 3.13.5, NumPy 2.3.5, SciPy 1.17.0, pandas 2.2.3, scikit-learn 1.8.0, and PyTorch 2.10.0+cpu on Linux x86_64. The virtualized processor identifies itself in

the recorded environment file. BLAS/OpenMP counts are set through both environment variables and `threadpoolctl`, because NumPy may already be imported before the driver starts. Actual thread-pool settings are recorded, and PyTorch is restricted to one thread. The artifacts include the configuration and source hashes. Numerical results can vary slightly with linear algebra libraries or hardware, particularly in near-tied candidate comparisons.

From the research-companion root, which contains `research/`, the unified commands are

```
python research/replicate.py -tables
python research/replicate.py -tests
python research/replicate.py -rerun -output
research/results_fresh
```

The first command regenerates manuscript assets from `results_current`. The last requires a new output directory, executes the current implementation for all protocols, then regenerates tables and figures. It does not use the embedded historical code or manuscript. The research-companion README documents individual drivers, dependency versions, result manifests, and the scope of numerical checks. The lightweight Overleaf package contains the manuscript assets, but does not run these commands by itself.

K Additional empirical results

K.1 Magnitude error versus ranking

Table 4 reports the percentage of queries whose maximum attribution error over 16 targets exceeds 10^{-3}, with 10^{-8} comparison slack. The identical aggregate failure rates of fixed and reselected IF in this table do not imply identical predictions or identical errors. Their magnitude and ranking behavior differ. A coarse failure indicator at a stringent threshold can hide those differences.

Table 4: Percentage of queries exceeding the 10^{-3} error tolerance. Entries are mean $\pm$ a 95% Student-t half-width across ten seeded splits. A zero half-width for SWA records the absence of observed failures, not a zero upper bound on failure probability in other problems.

Method	Breast cancer	Wine	Digits	Checkpoints
Fixed IF	81.5 ± 2.3	75.3 ± 1.4	99.4 ± 0.6	92.7 ± 1.2
Fixed exact	19.1 ± 9.1	13.0 ± 11.4	17.2 ± 7.8	12.3 ± 9.4
Reselect IF	81.5 ± 2.3	75.3 ± 1.4	99.4 ± 0.6	92.7 ± 1.2
Reselect Newton	77.2 ± 2.3	73.1 ± 2.0	94.8 ± 1.6	87.5 ± 1.2
IF + Newton	40.3 ± 9.3	45.0 ± 4.3	49.0 ± 5.9	45.5 ± 6.9
SWA	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0

Table 5: Mean Spearman correlation across groups, aggregated within each seed and then across ten seeds. Each ranking task fixes a dataset, split, deletion size, mechanism, and target. The last row is only the restricted linear score in (23), not the complete TRAK method.

Method	Breast cancer	Wine	Digits	Checkpoints
Fixed IF	0.740 ± 0.099	0.786 ± 0.127	0.868 ± 0.035	0.839 ± 0.108
Fixed exact	0.749 ± 0.112	0.834 ± 0.145	0.886 ± 0.036	0.854 ± 0.117
Reselect IF	0.850 ± 0.061	0.870 ± 0.040	0.900 ± 0.019	0.913 ± 0.029
Reselect Newton	0.933 ± 0.034	0.935 ± 0.020	0.970 ± 0.011	0.951 ± 0.015
IF + Newton	0.972 ± 0.026	0.958 ± 0.017	0.992 ± 0.003	0.987 ± 0.007
SWA	0.999 ± 0.001	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
Linear TRAK-style	0.378 ± 0.069	0.493 ± 0.105	0.788 ± 0.032	0.661 ± 0.079

IF + Newton attains mean ranking correlation of at least 0.958 in every setting while failing the absolute tolerance on at least 40.3% of queries. This is consistent with the distinction between ranking and magnitude already emphasized in group-influence work (Koh et al., 2019). The certified method

is not required to preserve an exact ordering between effects separated by less than its tolerance, so a correlation slightly below one is compatible with its point-error guarantee.

K.2 Conditioning on whether the selected configuration changes

Table 6 separates stable and switched queries. MAE is computed across the 16 coordinates for each query, then averaged across the queries in that stratum. Sign accuracy is computed within each query only for reference effects whose absolute value exceeds 10^{-8}, then averaged. These are pooled descriptive conditional means, not seed-weighted uncertainty intervals. A rounded MAE of zero need not be exactly zero.

Table 6: Conditional attribution errors and sign accuracy. Stable means the selected candidate is unchanged. Switched means the complete procedure selects a different candidate. MAE is in signed-logit units. A small absolute error does not certify the sign of an effect near zero, which explains the occasional sub-100% SWA sign accuracy.

Setting	Method	MAE stable	MAE switch	Sign stable (%)	Sign switch (%)
Breast cancer	Fixed IF	0.0322	3.1577	97.4	56.4
Breast cancer	Fixed exact	0.0000	3.2575	100.0	55.7
Breast cancer	Reselect IF	0.0406	2.2795	97.2	77.6
Breast cancer	Reselect Newton	0.0405	1.4318	97.7	91.1
Breast cancer	IF + Newton	0.0119	1.0825	99.7	95.2
Breast cancer	SWA	0.0000	0.0000	99.9	100.0
Wine	Fixed IF	0.2239	3.5456	95.5	57.7
Wine	Fixed exact	0.0000	3.3646	100.0	60.8
Wine	Reselect IF	0.2255	2.8210	95.5	77.1
Wine	Reselect Newton	0.1565	0.6543	97.1	96.4
Wine	IF + Newton	0.1058	0.4935	98.3	97.8
Wine	SWA	0.0000	0.0001	100.0	100.0
Digits	Fixed IF	0.0421	0.4811	96.7	65.0
Digits	Fixed exact	0.0000	0.4557	100.0	64.8
Digits	Reselect IF	0.0436	0.3781	96.5	76.1
Digits	Reselect Newton	0.0225	0.1199	97.5	93.3
Digits	IF + Newton	0.0043	0.0426	99.7	97.8
Digits	SWA	0.0000	0.0000	99.9	100.0
Frozen checkpoints	Fixed IF	0.0409	0.8702	96.6	56.2
Frozen checkpoints	Fixed exact	0.0000	0.9014	100.0	56.3
Frozen checkpoints	Reselect IF	0.0478	0.4399	96.3	82.2
Frozen checkpoints	Reselect Newton	0.0438	0.2002	96.6	92.8
Frozen checkpoints	IF + Newton	0.0075	0.0726	99.6	97.6
Frozen checkpoints	SWA	0.0000	0.0000	100.0	100.0

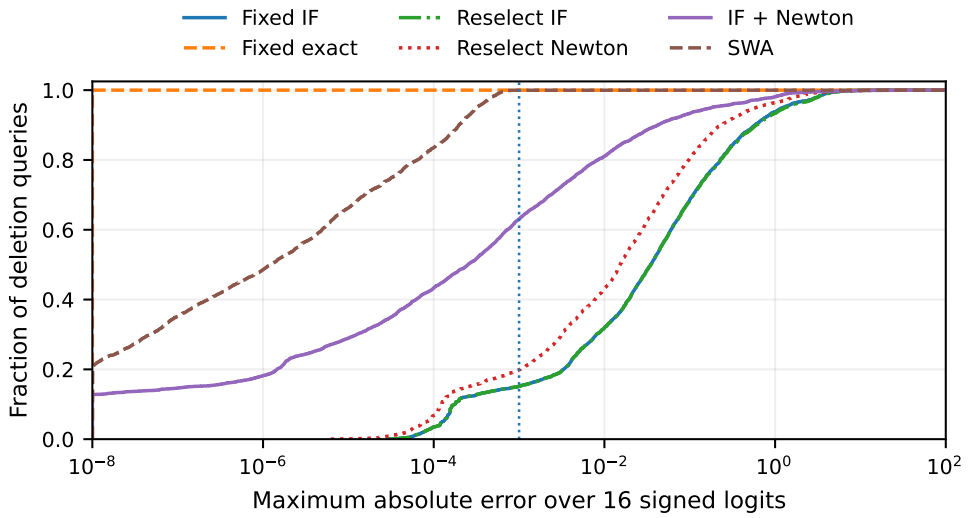


Figure 1: Error distribution conditional on a stable selected configuration. The numerical reference for *Fixed exact* is identical to the full reference in this stratum. Values below 10^{-8} are displayed at the left boundary. The dotted line is 10^{-3}.

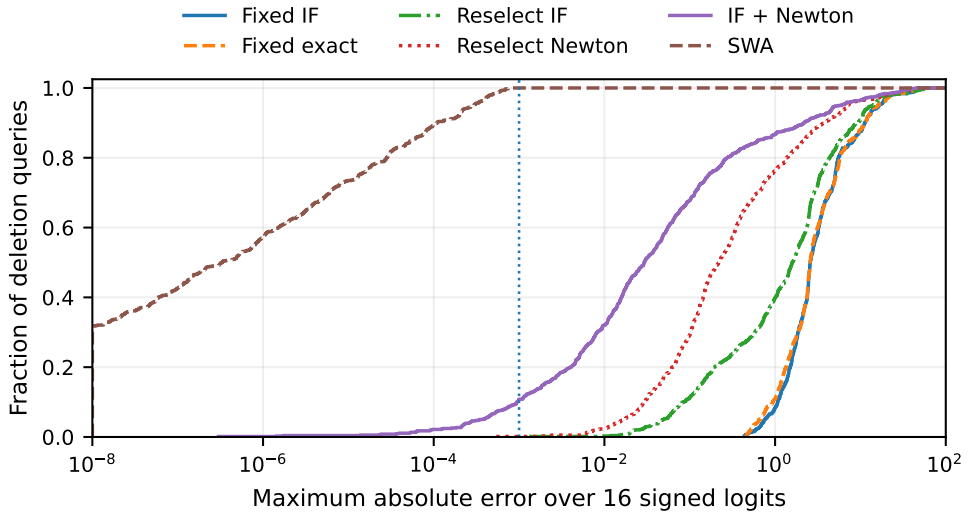


Figure 2: Error distribution conditional on a changed selected configuration. Exact retraining of the original winner need not approximate the selected output. Values below 10^{-8} are displayed at the left boundary. The dotted line is 10^{-3}.

K.3 Deletion mechanism and fraction

Table 7 preserves all six conditions rather than selecting high-switch groups. Its uncertainty is computed across the ten seed-level condition means. Seeded splits can produce very different original validation winners and different distances to selection boundaries. These half-widths quantify that descriptive variation, and are not binomial intervals based on independent deletion queries. The raw records support alternative weighting of interventions when a particular deployment distribution is specified.

Table 7: Results by deletion fraction and mechanism. Every row uses 20 groups per seed and ten seeds. All rates are percentages. No intervention condition was selected according to its realized reselection frequency.

Setting	Deleted	Mechanism	Reselection (%)	IF + Newton failure (%)
Breast cancer	1%	neighborhood	6.5 ± 4.8	11.0 ± 5.0
Breast cancer	1%	uniform	5.0 ± 3.4	13.5 ± 7.9
Breast cancer	5%	neighborhood	15.0 ± 7.9	32.5 ± 10.8
Breast cancer	5%	uniform	24.0 ± 14.8	52.5 ± 16.5
Breast cancer	10%	neighborhood	31.5 ± 13.0	60.5 ± 11.0
Breast cancer	10%	uniform	32.5 ± 19.7	72.0 ± 18.4
Wine	1%	neighborhood	2.0 ± 3.0	14.0 ± 7.5
Wine	1%	uniform	3.0 ± 2.5	13.5 ± 5.1
Wine	5%	neighborhood	12.5 ± 14.9	41.5 ± 11.1
Wine	5%	uniform	14.5 ± 13.7	55.5 ± 8.8
Wine	10%	neighborhood	24.0 ± 21.4	62.5 ± 11.8
Wine	10%	uniform	22.0 ± 18.9	83.0 ± 5.1
Digits	1%	neighborhood	1.0 ± 1.5	16.5 ± 7.4
Digits	1%	uniform	0.5 ± 1.1	4.0 ± 1.5
Digits	5%	neighborhood	22.5 ± 12.1	89.5 ± 5.7
Digits	5%	uniform	10.0 ± 8.8	30.5 ± 13.4
Digits	10%	neighborhood	48.5 ± 16.5	98.5 ± 1.7
Digits	10%	uniform	20.5 ± 14.7	55.0 ± 18.9
Frozen checkpoints	1%	neighborhood	4.5 ± 6.8	8.5 ± 3.8
Frozen checkpoints	1%	uniform	4.5 ± 7.2	6.5 ± 7.7
Frozen checkpoints	5%	neighborhood	13.0 ± 12.2	71.0 ± 8.6
Frozen checkpoints	5%	uniform	7.5 ± 11.2	29.5 ± 16.5
Frozen checkpoints	10%	neighborhood	31.0 ± 10.5	93.5 ± 2.9
Frozen checkpoints	10%	uniform	13.5 ± 15.2	64.0 ± 19.7

K.4 Matched certification and dense menus

Table 8: Mean main-query Newton refinements and the separate 600-query allocation control. Matched savings and speed-up divide pooled counts and elapsed time, respectively. Different reference and certificate accuracies make the first comparison a descriptive work ratio, not an equal-accuracy superiority result.

Setting	Reference steps	SWA steps	Matched savings (%)	Matched speed-up
Breast cancer	47.7	10.9	72.3	1.89
Wine	48.5	11.1	71.0	1.73
Digits	51.2	17.1	63.0	1.99
Frozen checkpoints	93.0	23.7	72.9	2.13

Table 9: Matched-certificate allocation ablation. Both policies require the same 10^{-3} maximum-coordinate output certificate. Uniform refinement continues updating all candidates. The savings entry is the mean $\pm$ a 95% Student-t half-width of five seed-level relative savings. This differs from the ratio of pooled counts used in Table 8. Times are mean milliseconds and are descriptive.

Setting	Queries	Uniform steps	SWA steps	Savings (%)	Uniform ms	SWA ms
Breast cancer	150	35.85	9.93	72.5 ± 4.0	9.26	4.91
Wine	150	35.97	10.44	70.8 ± 6.4	6.41	3.70
Digits	150	46.36	17.16	63.0 ± 1.8	35.05	17.59
Frozen checkpoints	150	87.10	23.59	72.8 ± 2.3	29.22	13.69

The matched ablation isolates candidate allocation. Every method computes the initial bank-wide approximation, and both allocation variants use the same certificate. The saving comes from avoiding

refinements of candidates already excluded as winners. It supports classical safe allocation for a selected-output query, rather than a new benefit from conditioning or output-based stopping.

Table 10: Dense-menu ablation, summing Newton refinements across both datasets, three seeds, two mechanisms, and five groups. Ambiguous counts runs that return a certified output with more than one surviving candidate. Each set of 60 deletion cases is reused across menus and tolerances. Rectangular removes cap conditioning, and Winner required also insists on a single survivor.

K	ε	Runs	Ambiguous	SWA	Rectangular	Winner required
65	0.001	60	0	3949	3949	3949
65	0.01	60	0	3949	3949	3949
65	0.1	60	1	3947	3947	3949
257	0.001	60	0	15413	15413	15413
257	0.01	60	0	15413	15413	15413
257	0.1	60	2	15409	15409	15413

The only ambiguous runs occur at $\varepsilon = 0.1$. There are three, each with two survivors. All concern digits, with identifiers $(r, K, \text{mechanism}, j)$ given by $(0, 257, \text{uniform}, 3)$, $(0, 257, \text{neighborhood}, 1)$, and $(2, 65, \text{neighborhood}, 0)$. Their maximum errors are approximately 0.03475, 0.03579, and 0.08472, respectively. Each saves two refinements relative to requiring the winner. Across all 360 runs, the output-stopping variants use 58,080 refinements, versus 58,086 when the winner is required. The cap-conditioned and rectangular variants have identical step counts throughout. The distinction is visible in these cases, but the additional savings from output agreement are negligible in this regime.

L NUMERICAL CHECKS, AUDITS, AND LIMITATIONS OF CERTIFICATION

L.1 IMPLEMENTED CHECKS

The script `test_theory.py` generates the numerical test configurations summarized in Table 11. Its random quadratic tests know the exact minimizer by a direct solve. The cap-support test compares the analytic support with a numerical constrained optimizer. Its discrepancy includes the numerical optimizer's own error. Tie tests enumerate the possible score arrangements used by the rectangular characterization. Ridge tests compare the group identity and truncation intervals with a direct retained-data solution. The counterexample is checked using Python rational arithmetic before converting values for display.

Table 11: Executed mathematical implementation checks. Counts describe tested configurations, not a substitute for the proofs. Values are rounded upward where appropriate.

Check	Cases	Largest observed discrepancy
Shifted-ball containment for strongly convex quadratics	1,000	No violation
Ball–halfspace support versus SLSQP	250	4.12×10^{-8}
Tie-aware feasible-winner enumeration	600	No failure
Ridge downdates and truncation intervals	500	6.67×10^{-16}
Bounded counterexample with exact rational arithmetic	7	Exact agreement

The main-result audit reads all 4,800 saved prediction records and verifies the stopping width, point errors, candidate retention, and interval containment against the saved tight references. The largest main interval width is 0.001997467 when rounded upward, below $2\varepsilon = 0.002$. The largest point error is below 0.000833410. The largest reference-center excess beyond an interval endpoint is below 1.057×10^{-10}, and no excess is larger than the 10^{-8} audit slack. The smallest main reference score gap is approximately 4.2312×10^{-7}. All original and main post-deletion selections also passed the implemented interval-separation assertion.

The dense ablation uses the same 10^{-8} containment slack and meets the requested point tolerance in every run. Every post-deletion reference winner is also interval-separated in the current dense run. The allocation ablation evaluates only computational work under the same mathematical stopping requirement, and is not counted as an additional independent accuracy experiment.

L.2 FLOATING-POINT BOUNDARIES

The implemented score and output bounds include a guard of $5 \times 10^{-13}(1 + |V(c)|)$. The same scale is used in the cap threshold. Near-empty cap constraints are clipped within the explicit floating-point tolerance in the code. These safeguards do not provide proven error bounds for every floating-point operation. In particular, they do not certify Cholesky backward error, the estimated validation spectral norm, or exponential-function rounding. The returned arrays therefore implement a real-arithmetic certificate numerically, rather than a formally verified certificate over all machine arithmetic.

A fully rigorous floating-point implementation would need outward-rounded evaluation of objective and gradient bounds, certified lower bounds on strong convexity, certified upper bounds on matrix norms, and explicit propagation of linear-system and transcendental-function error. It would also need verified comparisons for pruning and emptiness. None of these additional guarantees is claimed by the numerical checks above. Numerical failures in other environments must be reported rather than excluded from the results.

M OUTPUT TARGETS, DECISIONS, AND COMPUTATIONAL MECHANISMS

The experiments in this section use the same normalized objectives and the same certificate implementation as the main study. They address output agreement, the role of tolerance, predictive quality, and complete computational cost. The features remain fixed rather than being retrained end to end. Every declared parameter combination is reported, including zero-gain cases and unfavorable timing results.

M.1 WHAT IS INHERITED AND WHAT IS ESTABLISHED HERE

Table 12: Result-level positioning. The information characterization, optimization enclosures, and measured implementation costs are different levels of claim.

Work	Established ingredient or target	Distinction in the present results
Suzuki et al. (2014)	Suboptimal-model regions, prediction bounds, and validation exclusion	These bounds are inputs to the selected-output enclosure, not new screening geometry.
Xiang et al. (2016)	Ball–halfspace support, Lemma 3 and Eq. (27)	We apply the formula under a winning premise and track conditional information across refinements.
Wilson et al. (2020)	Deterministic guarantees for approximate cross-validation assessment and selection	Our estimand is the selected output after one specified deletion, rather than a cross-validation assessment functional.
Jamieson & Talwalkar (2016)	Allocation of computation to non-stochastic candidate identification	We permit an accurate output before identification. The finite-data separation compares specified residual-based races.
This paper	Tie-aware output set, conditional safety, disagreement-dependent work	Interval-information sharpness is not a lower bound against unrestricted data access; numerical work reductions are separately measured.

M.2 PREDICTIVE UTILITY AND THE DECOMPOSITION OF ATTRIBUTION

We evaluate the selected binary model on the complete held-out test split, not only the 16 queried examples. Table 13 reports accuracy and unpenalized binary log loss. Original quantities are averaged equally over ten original banks; retained quantities are averaged over the 120 queries per bank and then across the ten banks. The latter are descriptive repeated-intervention averages, not a new independent test sample. Validation score gaps are between the best and second-best candidate on each retained sample. The endpoint percentage counts a selected regularizer at either end of the declared grid, including within any checkpoint representation.

Table 13: Predictive quality of the selected original and retained models. Accuracy and endpoint selection are percentages. The gap is in mean validation-loss units.

Setting	Acc. before	Acc. after	Loss before	Loss after	Median gap	Edge (%)
Breast cancer	97.1	97.1	0.095	0.096	0.0017	0.9
Wine	98.6	98.7	0.031	0.034	0.00043	69.2
Digits	89.4	89.0	0.277	0.284	0.0016	0.4
Frozen checkpoints	93.7	93.5	0.185	0.187	0.0021	18.3

Table 13 shows that the models are predictive on these binary tasks, but their tuning behavior differs. Wine selects a regularization endpoint on 69.2% of retained queries, compared with 18.3% for frozen checkpoints. A switch therefore need not have the same interpretation on every split. These rates also do not establish that the grid brackets the optimum in every case. The raw records retain original and retained losses, norms, regularizers, and every candidate's validation loss and queried output.

For each coordinate we compute the decomposition in (3) from tightly fitted references. The within-configuration component uses the original winner refitted on the retained data. Subtracting that

component from the complete selected effect gives the reselection component. Table 14 restricts attention to switched queries and reports pooled coordinate quantiles. Its sign-change column compares the sign of the within-configuration effect with that of the total effect; it uses all coordinates rather than the 10^{-8} threshold of the earlier sign-accuracy table.

Table 14: Magnitude of the two attribution components on switched queries. Quantities are signed-logit effects, not percentage changes in predictive accuracy.

Switched queries	Median $\lvert\Delta^{\mathrm{fixed}}\rvert$	Median $\lvert\Delta^{\mathrm{sel}}\rvert$	P90 $\lvert\Delta^{\mathrm{sel}}\rvert$	Sign change (%)
Breast cancer	0.344	1.64	7.16	44.3
Wine	0.363	2.32	7.5	39.2
Digits	0.229	0.325	0.956	35.2
Frozen checkpoints	0.227	0.605	1.92	43.7

M.3 Complete probability-output design

The probability study uses breast cancer and digits with seeds $0, 1, 2$. Within each split it takes query identifiers

$$0, 1, 20, 21, 40, 41, 60, 61, 80, 81, 100, 101.$$

These are two groups from each fraction/mechanism condition of the main protocol. The 72 distinct deletion cases are reused with four menus, formed by $K \in \{19, 65\}$ and lower regularization endpoint 10^{-6} or 10^{-3}; the upper endpoint is always one. The grid is logarithmic. This gives 288 deletion/menu combinations. It does not condition group selection on a post-deletion score gap, model switch, output value, or work saving.

Each original and retained menu is tightly fitted for numerical evaluation. The reference winner must have an upper score bound strictly below every competitor's lower bound. A violation terminates execution. Every one of the 288 reference comparisons passes. The four refinement variants share the exact fitted original bank and influence starts. The conditional and rectangular variants differ only in prediction conditioning. The winner-required comparator uses rectangular output bounds and must additionally have one possible winner. The uniform comparator uses conditioning but refines all candidates. No reference fit is used to choose the online refinement schedule.

The probability target at signed feature $t = (2y - 1)x$ is

$$\sigma(t_{h_*}^\top \theta_{h_*}^-) - \sigma(t_{h_0}^\top \theta_{h_0}), \qquad \sigma(z) = (1 + e^{-z})^{-1}.$$

It is the change in probability assigned to the observed true label, not necessarily the probability of class one. Both endpoints of each signed-logit interval are transformed monotonically before baseline subtraction. In particular, the implementation never applies σ to a logit difference.

Trajectories are executed for the first of the original 16 targets and for all 16 jointly, at the tightest probability tolerance 10^{-4}. Runs at larger tolerances $10^{-3}, 10^{-2}, 0.05$ are exact prefixes of these trajectories because tolerance affects only stopping, not the next iterate. Reuse therefore gives the same refinement counts and endpoints as separate runs at those stopping times. It does not measure latency for the shorter runs.

We then evaluate *each* of the original 16 targets individually, rather than generalizing from the first target. No joint trajectory in this study eliminates a candidate through conditional infeasibility. Score pruning and candidate iterates are therefore identical under coordinate restriction, and an individual-target race is an exact prefix of the joint race. This yields all 4,608 case/menu/target combinations per tolerance. A separate verification reruns 24 fixed combinations in all four modes, or 96 direct races, with single-coordinate query matrices. Their counts agree exactly; the largest endpoint difference is 2.36×10^{-16}. The extraction script rejects reuse if a joint history has a conditional exclusion. This qualification matters because, in general, discarding a candidate on another coordinate can alter the schedule.

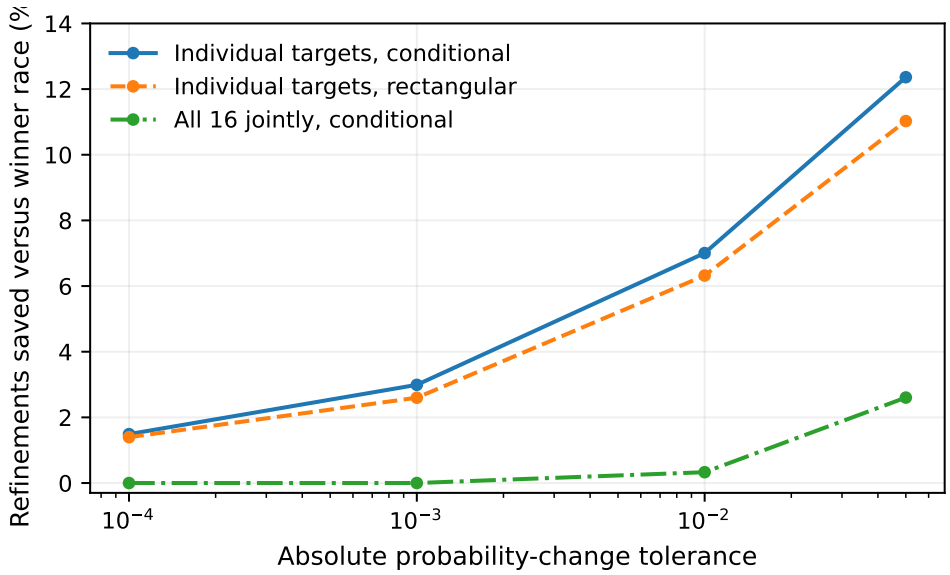


Figure 3: Refinement savings over equally certified winner identification, including every tested case. Individual-target curves average work across all 16 targets queried separately. The joint curves require the entire output vector to meet tolerance. Savings are ratios of pooled counts, not means of per-query percentages.

All 73,728 scalar method/tolerance records meet their requested probability point-error allowance. This count contains repeated targets, menus and tolerances; the independent randomization units are only the six dataset/split combinations. Table 15 therefore exposes every split's work ratio rather than treating the many coordinates as independent samples. Even at the loosest tolerance, most combinations save no refinements. The observed advantage is restricted to part of the tested regime, rather than a uniform acceleration guarantee.

Table 15: Percentage scalar refinement savings, conditional stopping versus winner-required stopping, for every split. Each entry includes all four menus, 12 groups, and 16 targets.

Setting	Seed	10^{-4}	10^{-3}	10^{-2}	0.05
Breast cancer	0	8.45	12.47	16.86	19.66
Breast cancer	1	4.32	6.76	11.62	16.67
Breast cancer	2	0.51	2.34	11.19	26.29
Digits	0	0.08	2.06	7.03	12.33
Digits	1	0.00	0.10	1.21	4.10
Digits	2	0.00	0.09	0.78	2.10

M.4 Sign and intervention-order decisions

A positive lower attribution endpoint certifies a positive effect, and a negative upper endpoint certifies a negative effect. An interval containing zero abstains. For two interventions with intervals $[a_G, b_G]$ and $[a_H, b_H]$, the difference belongs to $[a_G - b_H, b_G - a_H]$. Thus $a_G > b_H$ certifies that G has the larger effect. This conclusion does not require independent interventions. Conversely, overlapping marginal intervals need not establish that the order is truly unknowable, since joint information could be more informative.

Within each fixed dataset, split, menu, target, deletion fraction, and tolerance, the protocol has four deletion groups, two uniform and two neighborhood. We evaluate all six unordered pairs. There are 6,912 comparisons per tolerance, with no selection by the true difference. Table 16 gives reporting fractions and total counted pair work, taking each intervention to be queried separately. It does not credit computation that a specialized joint implementation might share. Reference comparisons use

the declared numerical allowance. No reported sign or order contradicts its reference beyond that allowance.

Table 16: Decisions supported by the returned scalar intervals. Sign coverage has 4,608 targets and ordering coverage has 6,912 intervention pairs at each tolerance. The same targets and deletions are reused across tolerances. Mean pair work sums the two separate query counts.

ε	Sign reports (%)	Pair-order reports (%)	Wrong orders	Mean pair work
0.0001	94.7	91.2	0	35.7
0.001	89.6	84.6	0	34.9
0.01	75.8	65.4	0	33.1
0.05	60.0	45.2	0	31.1

At $\varepsilon = 0.01$, 75.8% of signs and 65.4% of intervention orders are reported. These reporting fractions describe computational decisions on the observed data, not statistical confidence levels. A sign can remain unresolved even at a tiny tolerance when its effect is near zero. A larger tolerance can also resolve a sign immediately when the entire interval is far from zero. The table evaluates signs at magnitude-based stopping; it does not claim optimal sign-specific allocation.

M.5 A MATCHED-WORK ITERATIVE POINT COMPARATOR

At every recorded round, we evaluate each active candidate at its current iterate, choose the lowest observed validation loss with the same tie order, and return that candidate's point output. This uses the same retained-data refinements as the certifying method and is stronger than restricting an approximation to one step. Its choice uses evaluated losses, not the numerical-reference winner. The point comparator shares the certificate-driven elimination schedule. Its overhead therefore need not match that of a standalone approximate racing strategy.

Table 17 compares these point estimates with the interval midpoint at the certificate's stopping round. The single-output entries concern the explicitly executed first target; the joint entries use all 16. The point estimates meet every evaluated tolerance and typically have smaller median error. The interval justifies stopping, rather than reducing observed error at equal work. A point-only adaptive schedule might attain the same accuracy at lower cost.

Table 17: Point approximations at the same counted work as conditional certification. Each row has 288 deletion/menu combinations. The midpoint has zero observed tolerance failures in every row.

Outputs	ε	Point failures (%)	Median point error	Median midpoint error
1	0.0001	0.0	1.1e-08	2.3e-07
1	0.001	0.0	9.7e-08	2.8e-06
1	0.01	0.0	6.2e-07	1.3e-05
1	0.05	0.0	5e-06	0.00012
16	0.0001	0.0	2.2e-08	2.3e-07
16	0.001	0.0	3.3e-07	3.4e-06
16	0.01	0.0	1.9e-06	2.2e-05
16	0.05	0.0	1.2e-05	0.00026

Table 18: Joint 16-output trajectories under a round budget. Each row includes all 288 combinations, retaining the final state of already stopped runs. Failure and certification use probability tolerance 10^{-3}. A budgeted point estimate can meet the reference tolerance without its interval certifying that accuracy.

Round cap	Mean steps	Median point error	P90 error	Point failures (%)	Certified (%)
0	0.0	0.0031	0.048	72.2	8.7
1	11.6	1.1e-05	0.0015	11.5	51.4
2	16.6	1.6e-07	6.3e-06	2.8	78.8
3	18.0	2.7e-08	2.6e-06	0.0	95.1
4	18.2	2.4e-08	2.2e-06	0.0	99.3

M.6 Pre-refit screening and observed geometry

The pre-refit test evaluates the retained gradient at the original fitted parameters, then forms score and prediction regions without applying an influence update. The test evaluates a retained-data gradient for every candidate and uses validation data to propagate each region, so these costs must be counted. Table 19 contrasts its resolution frequency with that of the influence starts before any retained-objective Newton refinement. Single-output entries again concern the first target, while joint entries concern all 16. The same 288 deletion/menu combinations appear in every row. The frequencies describe sufficient resolution, not minimum total cost. The main algorithm uses the influence start without this optional additional pass.

Table 19: Resolution before retained-objective Newton refinement. Original-fit screening precedes the influence update; IF-start screening includes it. Neither percentage counts this bank-wide work as a free operation.

Outputs	ε	Original-fit resolution (%)	IF-start resolution (%)
1	0.0001	3.1	10.4
1	0.001	5.9	21.2
1	0.01	10.1	34.0
1	0.05	18.8	47.6
16	0.0001	0.0	4.9
16	0.001	0.0	8.7
16	0.01	0.0	19.4
16	0.05	5.2	38.9

For a nondegenerate cap, let $a = \nabla V(c)$, radius ρ, direction t, and best score upper bound u. The upper endpoint improves precisely when

$$\frac{t^\top a}{\|t\|\,\|a\|} > \frac{u - V(c)}{\rho\,\|a\|}.$$

The code records the guarded version of the right-hand side used in the numerical formula. Lower-endpoint improvement uses direction $-t$. We record both endpoint activity and accumulated probability hulls; counting an active endpoint is not counting a saved refinement.

For every joint trajectory and round, define b_k as the largest active score-interval width. We evaluate the reference-bank spread

$$\omega_k = \max_j \left\{ \max_{h: v_h - v_* \le 2b_k} \sigma(f_{hj}) - \min_{h: v_h - v_* \le 2b_k} \sigma(f_{hj}) \right\}.$$

This is a diagnostic computed from tightly fitted reference candidates, not an extra input to the online procedure. It corresponds to the near-optimal set in the work theorem. The recorded active spread instead uses only active candidates. It can be smaller because elimination retains more information than this single score-width envelope.

Table 20: Geometry along joint-output conditional trajectories. Later rows contain only runs that reach that round. The cosine entry is the median over runs of mean absolute output/validation-gradient alignment. "Tightened" is the pooled fraction of freshly evaluated endpoints improved by the cap.

Round	Runs	Ambiguous (%)	Median ω	Active spread	$\lvert\cos\rvert$	Tightened (%)
0	288	67.0	0.432	0.152	0.638	43.6
1	274	38.0	0	0	0.610	39.1
2	169	33.7	0	0	0.581	39.0
3	70	20.0	0	0	0.611	28.2
4	19	10.5	0	0	0.639	24.7
5	2	0.0	0	0	0.416	0.0

Many endpoints tighten without avoiding a complete synchronous round. Geometric activity alone therefore does not measure computational benefit. A different output may still determine the maximum width, or a discrete solver step may reach both stopping thresholds at once. The finite-data control below varies this latter feature directly. The margins, alignment, and disagreement are descriptive diagnostics. We do not fit them into a predictor of savings for deployment.

M.7 Finite-data mechanism and unfavorable controls

The complete grid uses the three-point sample of Proposition 6, deletes its third observation, and varies

$$\delta \in \{10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}\}, \quad \eta \in \{0, 0.01, 0.1, 1\}, \quad \varepsilon \in \{10^{-3}, 10^{-2}, 0.05\}.$$

The output direction is $(e_1 + \eta e_2)/\sqrt{1+\eta^2}$. Validation gap and output disagreement therefore vary independently without rescaling the query norm. Half-step and full-step Newton schedules are both evaluated. Four methods on these 120 configurations produce 480 method records. The $\delta = 10^{-2}$ control lies outside the proposition's sufficient range and is retained as an ordinary numerical case.

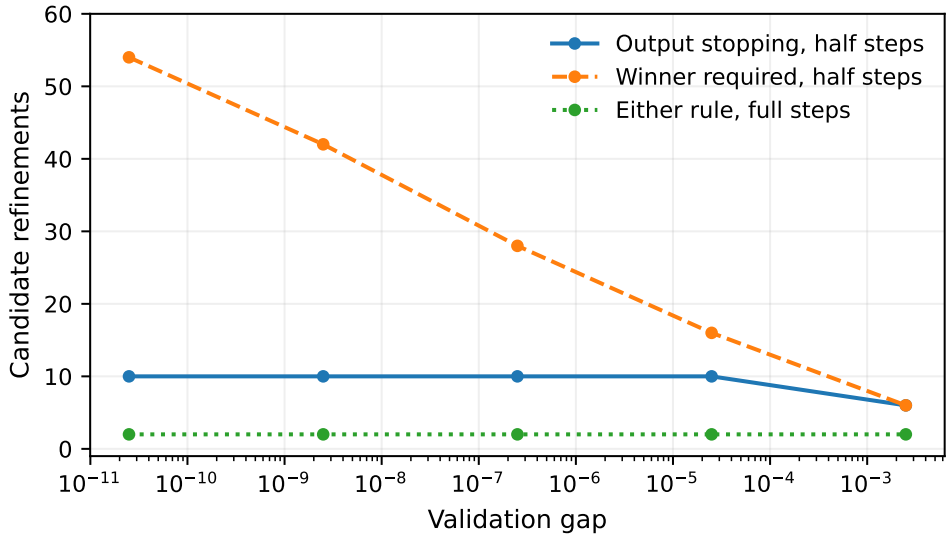


Figure 4: Actual residual-based races on the finite-data ridge construction, with aligned output and $\varepsilon = 10^{-3}$. Half-step refinement produces increasing winner-identification work as the validation gap decreases. Full Newton solves both candidates in one step each, eliminating the distinction. The direct quadratic solution is not a costly-retraining benchmark.

At $\delta = 10^{-8}$, $\eta = 0$, and $\varepsilon = 10^{-3}$, half-step output stopping needs ten refinements while the winner race needs 42. At the same gap with $\eta = 0.01$, both need 42 at this tolerance. The gap alone cannot explain that change. At looser tolerance some disagreeing cases again stop early. The conditional

and rectangular methods have identical counts throughout this mechanism grid. With full Newton all methods take two refinements. These controls distinguish a realizable certificate separation under a specified refinement rule from a practical recommendation to damp a quadratic solve.

M.8 COUNTERBALANCED COMPLETE-QUERY TIMING

We time breast cancer and digits with seeds $0, 1, 2$, query identifiers $0, 20, 40, 60, 80, 100$, the original 19-candidate grid, and probability tolerances $10^{-3}, 10^{-2}, 0.05$. Target choices are individual indices $0, 5, 10, 15$ and the joint set of 16. Four cyclic method orders are executed for each configuration, rotating their initial offset across queries. Every method occupies every execution position 540 times. The 8,640 calls are direct executions, not durations assigned to saved trajectory prefixes.

The timer includes constructing the retained mask and shared retained arrays, computing every influence start, constructing candidate states, all score/output certificates, pruning, and Newton refinements. The original bank is cached. No reference fitting or reference-accuracy audit is charged to any query method. Preprocessing plus complete bank time, and the narrower fitted-parameter/factorization bank time, are recorded separately. This design measures complete implementations of the four compared races on one CPU environment; it does not establish a cross-machine speed ordering.

Table 21: Counterbalanced query timing. Millisecond columns are pooled medians. Ratio columns divide pooled elapsed time, rather than dividing those medians. Scalar pools the four fixed individual target indices. Full quartiles and paired records accompany the table.

Output	ε	Cond. ms	Rect. ms	Winner ms	Uniform ms	Cond./win.	Rect./win.
Joint16	0.001	8.11	6.36	6.51	14.92	1.129	0.997
Joint16	0.01	8.21	6.76	6.88	14.79	1.121	0.998
Joint16	0.05	8.01	6.57	6.41	14.89	1.119	0.989
Scalar	0.001	7.55	6.19	6.46	14.43	1.092	0.981
Scalar	0.01	7.51	6.20	6.63	13.80	1.078	0.976
Scalar	0.05	7.43	6.10	6.31	13.56	1.053	0.951

Conditioning is slower than winner identification in these pooled comparisons. Rectangular output stopping uses 95.1–98.1% of the winner race's time for scalar queries and 98.9–99.8% for joint queries. These modest differences do not support a broad deployment-speed claim. The cap's small refinement saving does not offset its evaluation cost. Uniform refinement remains substantially more expensive, consistent with the established value of candidate elimination.

Output and winner races share the same existing candidate bank, so their comparison does not require duplicate bank construction. For a different comparison that constructs a bank only for this method, a break-even query count would require a matched competitor without that bank. No such break-even advantage is inferred from unrelated timer scopes. Memory records count logical cached arrays, not peak process residency.

M.9 CONTROLLED SAMPLE-SIZE SCALING

We generate 30,000 64-dimensional Gaussian feature vectors. Consecutive coordinates follow an AR(1) construction with correlation 0.5. The binary probability is the sigmoid of a linear score on the first eight coordinates, with coefficients $1.1(1, -1, 1, 1, -1, 1, -1, 1)$. A source identifier is uniform on four values. In the first 20,000 records, source-zero labels are flipped independently with probability 0.3. The validation and test pools, 5,000 records each, are unflipped. Seeds are $22000 + r$, $r \in \{0, 1, 2\}$.

Training uses the first 2,000 or 20,000 records, so the smaller design is a prefix of the larger one. Each size has its own original-training standardization, divided by eight, followed by a penalized intercept. The menu is 17 values logarithmically spaced in $[10^{-6}, 1]$. We delete $1\%, 5\%, 10\%$ uniformly from training or uniformly from source zero. The generator seed is $33000 + 1000r + n + \mathrm{round}(100f)$. All 36 size/seed/fraction/mechanism cases are evaluated with the four methods and three probability tolerances, producing 432 records on the same 16 test targets. Every original and post-deletion reference passes score separation, and all point errors meet tolerance.

Table 22: Controlled-simulation predictive quality and logical storage. Test metrics average over three seeds. The condition range is for the selected original Hessians. Decimal MB counts feature arrays and cached bank parameters/factors separately, not peak process memory.

n	Accuracy (%)	Log loss	Brier	Hessian cond.	Data MB	Bank MB
2000	77.73	0.4672	0.1527	278.7–312.2	6.25	0.583
20000	79.46	0.4479	0.1444	278.4–282.0	15.61	0.583

Table 23: Total refinements in the controlled scaling experiment. Each row sums 18 deletion cases, three seeds times six intervention conditions. No favorable cases are omitted.

n	ε	Conditional	Rectangular	Winner	Uniform
2000	0.0001	210	210	210	646
2000	0.001	208	208	208	612
2000	0.01	196	196	208	510
20000	0.0001	139	139	139	561
20000	0.001	133	133	133	459
20000	0.01	130	130	130	408

Every original bank selects $\lambda = 10^{-6}$. The calculation measures predictive accuracy and sample-size scaling, but provides a limited test of interior model reselection. At 20,000 training observations the three adaptive variants again have identical counts. The uniform comparison measures savings from elimination only. The experiment is entirely simulated, and increasing its sample count does not substitute for testing an additional external dataset or a high-dimensional pretrained representation.

A separate four-order timing run covers all three seeds, both training sizes, both 5% deletion mechanisms, and $\varepsilon = 10^{-3}$, for 192 complete calls. It uses the same timer scope as the smaller timing study and separately records feature preparation and complete bank construction. The source and raw timings permit amortization under a chosen query workload; no universal amortization threshold is asserted.

Table 24: Controlled-simulation complete-query timing at probability tolerance 10^{-3}. Each query-method median pools 24 directly executed calls at that training size. Bank and data-preparation medians use three seeds and are separate measurements, not additive latency quantiles.

n	Cond. ms	Rect. ms	Winner ms	Uniform ms	Bank ms	Data ms
2000	23.84	22.93	22.44	49.56	84.8	74.5
20000	116.30	121.59	110.01	296.38	989.1	59.8

M.10 Versioned execution and verification

Every driver imports `research/code/core.py`. Tables and figures read `research/results_current`. Historical records remain separate. Fresh main and dense runs match all 4,800 original refinement counts and switch indicators and all 360 dense conditional counts, including the exceptional early stops. Current tables use the fresh records, not this agreement as an assumption.

Regression tests cover ordered ties, duplicates, unconditional contradictions, conditional exclusion, probability transformation, and unresolved returns. The quadratic in Appendix H excludes the losing candidate after one refinement. Original property tests, rational ridge identities, complete main-record checks, and direct coordinate-prefix reruns also pass. These are numerical and algebraic checks, not formal proof or floating-point verification.